The Dissertation Committee for Todd Ammon Thornock certifies that this is the approved version of the following dissertation:

How the Timing of Performance Feedback Impacts Incentive-Based Individual Performance

Committee:

Steven J. Kachelmeier, Co-Supervisor

Michael G. Williamson, Co-Supervisor

Urton L. Anderson

Volker Laux

Caroline A. Bartel

How the Timing of Performance Feedback Impacts Incentive-Based Individual Performance

by

Todd Ammon Thornock, B.S.; M.Acc.; M.S.Acc.

Dissertation

Presented to the Faculty of the Graduate School of

the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August, 2011

Dedication

For my eternal companion, Samara Elizabeth Thornock, for her tireless support.

For my children, Spencer, Mirah, Jonah, Malcolm, and Quinton, for enduring a student father.

Acknowledgements

I appreciate the excellent guidance of my co-chairs, Steve Kachelmeier and Michael Williamson, and of the members of my dissertation committee, Urton Anderson, Volker Laux, and Caroline Bartel. I will always be in debt to Steve and Michael for taking a chance to mentor and teach me. I am especially grateful for Michael's availability and willingness to answer my questions and calm my worries and for Steve's readiness to share his experience in the world of academia.

I gratefully acknowledge support from the Institute of Management Accountants Doctoral Research Grant and the Eugene and Dora Bonham Research Fund.

Lastly, I am grateful for my friends in the doctoral program who have gone before me and come after me for the collegial environment to work and grow. I am particularly grateful for Marcy Shepardson who has always been a support and a friend. I am also grateful for James Smith who has always been there to chat with about life and act as a sounding board for my ideas.

How the Timing of Performance Feedback Impacts Incentive-Based Individual Performance

Todd Ammon Thornock, Ph.D.

The University of Texas at Austin, 2011

Supervisors: Steven J. Kachelmeier and Michael G. Williamson

Performance feedback plays an important role in management accounting, as it is integral to performance measurement and evaluation. The *timing* of performance feedback is a critical characteristic of accounting information systems and is often a choice variable for managers and management accountants. In this dissertation, I examine the relation between the timing of outcome-based performance feedback and individual performance.

I find that immediate outcome-based performance feedback, while benefiting current performance, can limit individuals' propensity to seek learning opportunities,

reducing future performance. Further, I find that feedback given after intermediate delays benefits future performance with a small cost to current performance. Lastly, feedback given after too long of a delay not only limits current performance, but also limits future performance due to the effects of information overload. Overall, I find support for an inverted-U relation between the timing of performance feedback and future performance.

In a two-period setting in which the timing of outcome-based performance feedback is manipulated in the first period and feedback is unavailable in the second period, I find that participants given intermediate feedback perform significantly better in the second period than those given feedback either after no delay or after a long delay.

I also investigate the processes by which performance is affected by the timing of performance feedback. These results contribute to a better understanding of the effect of performance feedback timing in complex task environments and provide insight into how delays in performance feedback can benefit or harm future performance.

Table of Contents

List of Tables and Figures ix

Chapter 1: Introduction 1

Chapter 2: Research Setting 8

Chapter 3: Literature Review and Hypothesis Development 13

Chapter 4: Maze Experiment 24

Chapter 5: Conclusion 49

Appendix A: Experimental Materials 52

Appendix B: Visual Basic Code for Maze Task 63

Bibliography 127

Vita 131

List of Tables and Figures

Table 1 42
Table 2 43
Table 3 44
Table 4 45
Table 5 46
Table 6 47
Table 7 48

Figure 1 12
Figure 2 23
Figure 3 39
Figure 4 40
Figure 5 41

Chapter 1: Introduction

Complex or ill-structured tasks are common in many organizational environments (e.g., sales representatives interact with customers, computer programmers debug code, audit associates perform analytical procedures, academics write research papers, etc.) (Simon, 1973; Kirsch, 1996). As such, effectively training individuals to perform complex tasks at a high level is in the best interest of these organizations. During this training, managers face the dilemma of how long to allow the individual to struggle to learn the task before providing outcome-based performance feedback.[1] If feedback is given too soon, individuals could use it to complete the task quickly now without taking advantage of the feedback to understand how to perform the task better in the future. If feedback is given too late, individuals could be less able to use the information in the feedback to improve performance in the task. This study seeks to shed light on the effect of outcome-based performance feedback timing on individual incentive-based performance in a complex task setting, particularly its effect on future performance.

This research question is important for a number of reasons. First, providing outcome-based performance feedback to decision makers within an organization is a critical element of performance measurement and evaluation and is a major objective for management accountants (Luckett and Eggleton, 1991; Bonner and Sprinkle, 2002; Sprinkle and Williamson, 2007; Hannan et al., 2008). One characteristic that presents managers and management accountants with an important choice is *when* to

[1] Outcome-based performance feedback is defined as information about the realized state(s) of nature from prior decisions or actions (Frederickson et al., 1999).

communicate outcome-based feedback to the decision maker. Prior research in accounting on the communication of feedback to individuals has investigated the effect on individuals of withholding outcome-based performance feedback (Ray, 2007), but not the effect of delaying this information.

Second, research outside of accounting on the effect of feedback timing in complex task settings has generally shown that delays in feedback *hurt* performance and learning (Lewis and Anderson, 1985; Diehl and Sterman, 1995; Brehmer, 1995; Gibson, 2000). This research has generally conceptualized feedback delay as just a time delay before receiving feedback and has not taken into consideration that, in complex settings, individuals can learn more information about the task as a result of a delay in feedback. In essence, feedback delay in study is less about the actual passage of time and more about the phase in the task in which feedback is given. For example, in most real-world environments, if there is a delay before receiving outcome feedback, an individual must still continue to work. By continuing to work on the task, the individual will gather more information about the task (i.e., task information)[2] in the intervening time before receiving feedback. Under this condition of task information increasing with feedback delay, research in the information load literature (Jacoby et al., 1974; Shields, 1983) suggests that performance and learning could potentially *improve*. However, this research also concludes that too much information can limit performance.

[2] The term "task information" in this paper refers to information inherently available within a task, such as task structure, task environment, etc. that can help the individual perform the task better.

The setting I employ in this study represents an important one found in many organizational environments. In this setting, an individual performs a complex task for a performance-based wage in each of two distinct periods. In the first period (i.e., current or training period), outcome-based performance feedback on the correctness or incorrectness of the individual's actions is available, while in the second period (i.e., future period), this feedback is no longer available. This setting is of interest to managers and management accountants because it mirrors many real-world settings that consist of both a formative, training period in which performance feedback is present as well as a period that limits or precludes the availability of feedback, in which performance is dependent on learning that has already taken place.

This task environment introduces a tradeoff to individuals regarding the use of feedback to affect current and future performance. An individual can choose to rely on the feedback to maximize efficiency in the shorter, current period (i.e., by failing to obtain sufficient task information to understand *why* the feedback was what it was) or to use the feedback to learn the task better in the current period (i.e., by obtaining sufficient task information to understand *why* the feedback was what it was). The former action will result in increased current performance but will likely lead to lower performance in the longer, future period when feedback is unavailable and the ability to learn significantly reduced (Salmoni et al., 1984). The latter action will likely positively affect future performance, though at some expense to current performance.

In this setting, how does the timing of outcome-based performance feedback in the current period affect individuals' future performance in the task? I posit an inverted-U relation between the timing of feedback and future performance, in which individual future performance increases in the delay of feedback to a point and decreases in additional delay of feedback. In a setting in which total task information available is held constant by allowing individuals to search for additional task information after receiving feedback, the receipt of outcome-based feedback may steer inherently myopic individuals away from the costly search for additional task information needed to learn (Thaler and Shefrin, 1981; Gabaix et al., 2006). The myopic decision to not seek additional task information would especially hinder the future performance of individuals with insufficient task information to learn at the point when feedback is given. In other words, if individuals receive immediate feedback at a point when there is insufficient information to learn, their myopic behavior of failing to seek needed additional task information will negatively affect learning and future performance.

A delay in the receipt of feedback would cause individuals to be exposed to more task information before feedback is given, allowing for sufficient task information to be available to learn and improve future performance. However, if feedback is given after too long of a delay, too much task information may be available to the individual, resulting in information overload effects that limit learning and future performance (Casey, 1980; Shields, 1983; Iselin, 1988).

I use an experimental task to test how the timing of outcome-based performance feedback influences individual performance. Participants complete a series of mazes in each of two periods and are paid a performance-based wage. All mazes contain visual cues that can help participants identify the correct and incorrect paths in both the first and second periods. In the first period, outcome feedback is provided either immediately (i.e., after no delay), after a short-intermediate delay, after a long-intermediate delay, or after a long delay. In the second period, this feedback is no longer available. All participants are informed that feedback would not be available in the second period and that the cues contained in the first period mazes would be consistent with the mazes of the second period. The participants are also informed that the second period includes twice as many mazes as in the first period (resulting in twice the potential payout for second period performance than for first period performance), emphasizing the real-world notion that future payouts from learning during the training period are substantial.

I find statistically significant support for the hypothesized inverted-U relation between the timing of outcome-based performance feedback and future performance. Specifically, I find that participants given feedback after no delay in the first period perform significantly worse in the second period than those given feedback after intermediate delays. This performance difference is associated with the myopic behavior of failing to search for additional task information needed to learn the cue patterns contained in the mazes. Also, I find that participants given feedback after a long delay in the first period perform worse in the second period than those given feedback after

intermediate delays. Additional analyses support the interpretation that this performance difference was attributable to the higher levels of task information facing participants given feedback after a long delay, with information overload hindering performance among those experiencing excessive delay.

This study contributes to the performance feedback literature by examining two process variables that affect the use of outcome-based performance feedback as well as the ability of individuals to learn from feedback – intertemporal tradeoffs (i.e., the tradeoff between current and future performance) and information overload (i.e., the inability to process information beyond a certain point). Understanding factors that affect individuals' use of performance feedback can better help to improve its use in organizations and in understanding specific environments in which delayed performance feedback is harmful or helpful (Libby and Luft, 1993; Luft and Shields, 2010).

Specifically, these findings can inform managers and managerial accountants on the potential benefits and pitfalls of delaying performance feedback during the training of individuals in complex task settings, an action management can often control. One implication is that the timing of performance feedback can influence learning processes. Specifically, outcome-based performance feedback given before sufficient task information is available for learning can amplify individuals' incentive to make myopic decisions that focus on current performance at the expense of future performance. As such, the importance of a delay of performance feedback should increase as management's concern for myopic employee behavior increases. Another implication

from this study is that too long of a delay in disseminating performance feedback can be detrimental to performance.

The remainder of this dissertation is organized as follows: Chapter 2 describes the research setting to examine the effect of feedback timing on individual performance. Chapter 3 reviews the literature relating to intertemporal tradeoffs and information overload and develops hypotheses stemming from these literatures. Chapter 4 presents the experimental task and results. In chapter 5, I conclude and provide suggestions for future research.

Chapter 2: Research Setting

The setting in which I examine the effect of outcome-based performance feedback timing is of particular importance to managers and management accountants. In this setting, an individual performs a complex task[3] during two periods. In the first, or training period, feedback regarding the correctness or incorrectness of the individual's actions is available to individuals after a certain interval of delay in the phase of the task, not just a time delay. In the second period, feedback is no longer available and the individual must perform the task based on what was learned in the first period. Real-world examples of a period in which performance feedback is no longer available (or even limited) include situations in which the training period on the task has ended, the source of the feedback is not available, the individual becomes one who is expected to *give* feedback, the test commences, etc. Brewster (2011) notes that, in many audit settings, performance feedback is often unavailable.

In this setting, the second period is designed to be longer than the first period. This characteristic is relevant to real-world settings, as it represents the notion that, while performance during the current or training period is important, future performance (i.e., after the training period) is more important to the long-term benefit of both the firm and the individual.

[3] Complex tasks are defined as tasks in which the state of the task changes due to both individual's actions as well as actions of nature and also require a series of interdependent decisions (Brehmer, 1995). Examples of common complex tasks that mirror the setting in this study are interacting with a customer, performing analytical procedures, writing a dissertation, doing math homework, etc.

To investigate the effect of outcome-based performance feedback timing on future performance, it is important to note what is represented by a delay in feedback. Prior research on the effect of the timing of feedback in complex tasks has failed to consider the potential for the level of task information (i.e., information about task structure, task environment, etc.) to increase concurrently with a delay of the phase of the task in which outcome-based feedback is provided (Lewis and Anderson, 1985; Diehl and Sterman, 1995; Brehmer, 1995; Gibson, 2000). In the many real-world scenarios that mirror this setting, individuals must continue to work on the task at hand while waiting to receive feedback. During this interval, individuals are obligated to receive additional information about the task *before* receiving feedback.[4] Individuals with less of a delay can also obtain this additional information, but must seek this information *after* receiving feedback. This additional task information, however acquired, can allow the individual to better incorporate the information contained in the feedback to improve task performance (Balzer et al., 1989). The notion of an increasing level of task information with an increase in feedback delay is illustrated in Figure 1, denoted by the solid line. The idea that *all* individuals have the same level of task information available to them is denoted by the dashed line in Figure 1.

This task setting allows for individual learning to occur. The task contains visual cues that indicate to the individual whether or not s/he is on the correct or incorrect path

[4] Actual measured time (e.g., milliseconds, minutes, days, etc.) may differ between the levels of delay, but the effect of the time element only has not been shown to have a clear effect one way or the other on individual performance (Salmoni et al., 1984; Kulik and Kulik, 1988; Schmidt, 1991; Butler et al., 2007). In this study, the actual measured time delay had a significant, low magnitude (compared to the task information proxy) effect on current performance and an insignificant effect on future performance.

to performing the task well. There is a minimum level of task information that individuals need to acquire in order to learn these cues, represented by the dotted line in Figure 1. Learning these cues will benefit individual performance both when feedback is available *and* when it is not available. Discovery of these cues is facilitated in the first (or training) period due to the presence of feedback and is extremely difficult in the absence of feedback (i.e., during the second period). The individual is charged with performing the task as best as possible in both periods and is aware of the need to learn the cues to ensure high future performance.

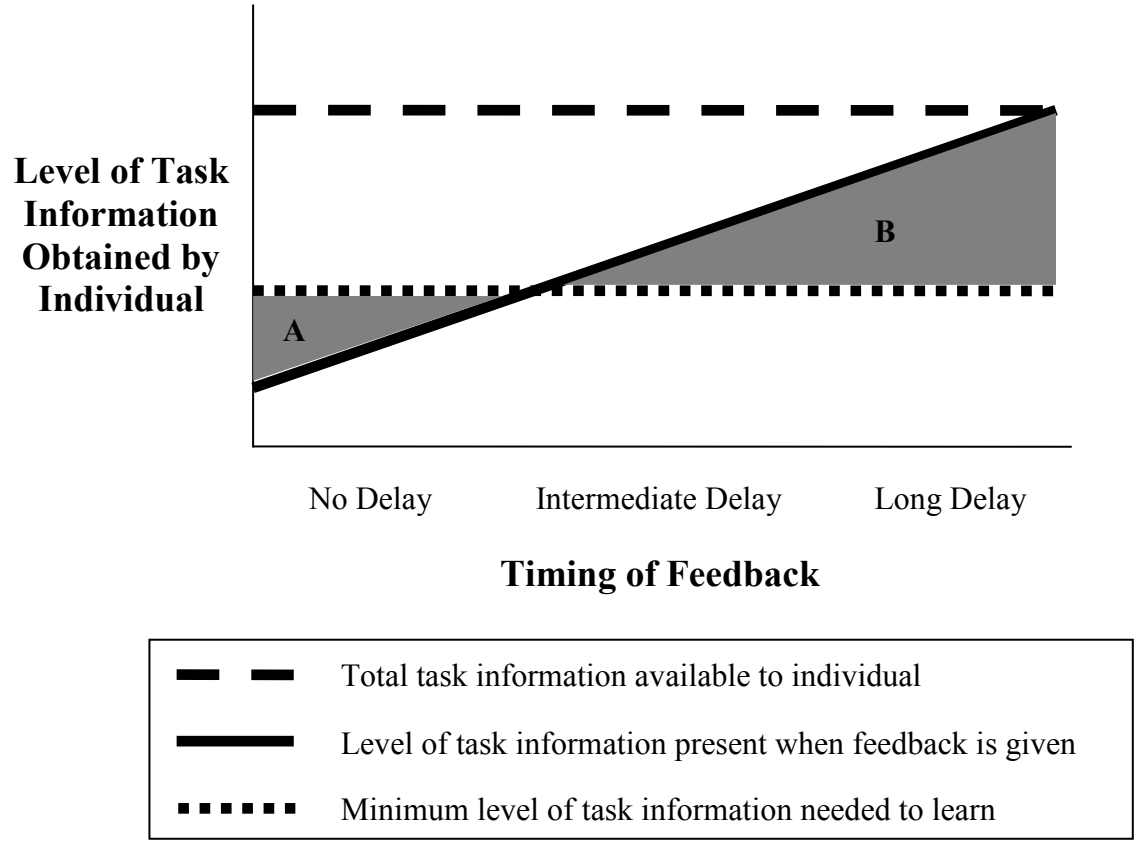
A key feature of this setting is that all participants are allowed to gather additional task information after outcome-based performance feedback is given and are instructed that this additional task information may be necessary for learning and improved future performance. In this study, the total level of task information available to individuals is held constant (see the dashed line in Figure 1). Accordingly information acquisition is, in part, exogenously determined by the extent of feedback delay but ultimately *endogenously* determined by the individual's choices to seek additional task information.

This study examines the effect of the timing of *outcome-based* performance feedback on individual performance. Outcome-based performance feedback is important in an organizational setting for a number of reasons. First, outcome feedback is often more visible, more unambiguous, and more readily available to managers and management accountants than is process feedback (Einhorn and Hogarth, 1981). Also, outcome-based feedback is often the only type of performance feedback available to

individuals due to the nature of the task or the time constraints inherent within the task (Bonner and Walker, 1994). Lastly, in complex task learning settings, process feedback is generally more costly to provide to individuals than is outcome feedback, due to the ill-structured nature of the task (Simon, 1973).

It is important to note why, in this study, process feedback cannot be instantly communicated to the individual on how to learn the cues in the task (cues representing signs that either indicate incorrect or correct ways of proceeding in the task). This design choice was made to represent a characteristic found in many complex tasks. First, in ill-structured or complex tasks, the cues are often not known with certainty and, therefore, cannot be communicated (Simon, 1973). For example, while an experienced teacher could give some tips (i.e., cues) to a new teacher on how to effectively teach a class, whether or not these tips (process feedback) will result in a successful class depends on so many factors (e.g., class size, type of students, time of day taught, subject matter, etc.) that the new teacher needs to actually teach the class to see if the tips are effective or not (outcome feedback). Second, even if the cues are known with certainty, it may be too costly to communicate them as they can represent specific knowledge, which is more costly to communicate that general knowledge (Jensen, 2001).

FIGURE 1
Graphical Illustration of Relation Between Task Information Level and Delay of Feedback



Chapter 3: Literature Review and Hypothesis Development

The tensions of intertemporal choice and information overload are key to understanding the effect that the timing of feedback has on individual performance. The purposes of this chapter are to 1) review relevant literature on intertemporal choice and information overload and to 2) develop the inverted-U hypothesis based on these literatures.

3.1 Literature Review of Intertemporal Tradeoffs

3.1.1 Tradeoffs in Research Setting

The two-period setting introduces an intertemporal tradeoff for individuals, particularly those who receive the feedback after less of a delay. On one hand, individuals can seek additional, costly information about the task during the first (i.e., training) period to better learn how to perform the task without feedback, an action that can benefit future performance. On the other hand, individuals can choose *not* to seek additional, costly information during the first period, an action that will benefit current performance at the expense of future performance. For example, if a PhD student receives immediate outcome-based feedback that the introduction to a research paper is "incorrect", s/he can either devote costly effort to understanding *why* the introduction was not up to par (e.g., by looking at well-written introductions in other papers, reading books on logical/persuasive writing, etc.) or just correct the introduction based on the feedback provided. The former action is likely teach the student how to write better introductions

in the future, while the latter action is unlikely to substantially benefit the student's future introductions.

The key tension that individuals face with respect to the intertemporal tradeoff in this setting is whether or not to pursue costly additional information about the task *after* the receipt of outcome-based performance feedback. Pursuing additional task information after receiving feedback is costly to the individual, in that this pursuit requires additional effort or time and/or lowers monetary payoffs. In the setting of this study, the costly search for additional task information after receiving feedback can represent an investment in the current period with the expectation of a greater return in the future period (i.e., higher future performance). However, individuals face uncertainty that this search for additional task information will result in the learning of the cues because learning the cues in the task is dependent on other factors in addition to the choice to seek additional information (e.g., individual skill, mental alertness, luck, etc.).

3.1.2 Differential Treatment of Current and Future Rewards

Many studies in the intertemporal choice literature have investigated the differential treatment individuals place on current versus future rewards. Anderhub et al. (2001) investigate the correlation between individuals' risk attitudes (i.e., level of risk aversion) and time preferences. They show that risk-averse agents tend to more highly discount future rewards than do risk-neutral or risk-seeking individuals, which supports the hyperbolic discounting notion of the tradeoff between current and future rewards. Individuals seem to place a psychological premium on current rewards over future

rewards due to risk aversion. Similarly, Lowenstein and Prelec (1992) analytically show that individuals employ nonconstant (i.e., hyperbolic) discounting when evaluating current and future rewards. Interestingly, these authors note that individuals often come to this cognitive realization while "succumbing to the impulse to consume." This concept of future rewards being discounted versus current rewards is applicable to many situations, including the potential effect of feedback timing on current and future performance.

From these studies, the concept of uncertainty of future rewards seems to take a prominent role in intertemporal choices. Keren and Roelofsma (1995) examine the relationship between time delay (intertemporal decisions) and uncertainty (risky decisions) by looking at two related effects – immediacy and certainty. The immediacy effect is the tendency for decision makers to elevate the significance of immediately experienced outcomes compared to delayed outcomes. One example of this effect is an individual's desire to have something immediately instead of in the future, as displayed in impulse buying. The certainty effect relates to the observation that individuals tend to overweight outcomes considered to be certain compared to outcomes that are merely probable (Kahneman and Tversky, 1979). An example of this effect is an individual's desire to take the sure thing instead of a chance at something much greater, since there is risk that the latter will not be realized. Keren and Roelofsma (1995) tentatively conclude that the immediacy effect derives from the certainty effect, but not vice versa. This

implies that individuals will tend to overweight current rewards to future rewards primarily due to the uncertainty of future rewards.

The implication of the above discussion for the research question of this study is that individuals have a tendency to focus on the near-term rewards of not seeking additional, costly information instead of a long-term focus learning and higher future performance due to the uncertainty that seeking additional task information will actually result in learning the cues, which would increase future performance and payouts.

3.2 Literature Review of Information Overload

3.2.1 Varying Task Information Levels in Research Setting

The information environment of this setting is relevant to understanding how feedback timing affects performance. As discussed in Chapter 2, increases in the level of feedback delay obligate individuals to be exposed to additional levels of task information (see the solid line in Figure 1). However, the total level of task information available is held constant for all individuals (denoted by the dashed line in Figure 1), ultimately determined by the individuals' search for additional task information *after* receiving feedback. This feature of the task causes the level of task information actually acquired (i.e., information load) ultimately to be an endogenous choice by the individual. The level of information load facing decision makers is a relevant issue in ill-structured, complex task settings as individuals have limited cognitive capacities to process information (Miller, 1956; Hertwig and Todd, 2003).

3.2.2 Effects of Information Load on Performance

Many studies have examined the effect of information load on individual performance. The notion that some information helps improve individual performance and that too much hurts performance was investigated in some depth by Jacoby and associates, particularly in the arena of consumer behavior. In Jacoby et al. (1974), the authors find that individuals' performance (as measured by their accuracy in brand selection) increases in the level of information load to a point and then decreases. They attribute negative effect of additional information beyond a certain point to be caused by the individuals' finite ability to process information in a given period of time.

Research in information load and overload is of interest to accountants, given the integral relationship between accounting and information (Balakrishnan et al., 2009). Shields (1980) examines the effect of information load on managers' analysis of reports. He shows that information load seems to affect search patterns and provides some theoretical support for information load affecting manager's decision making. In a follow-up study, Shields (1983) investigates the effect of information load on judgment accuracy. The author examines this research question in a setting with corporate managers and their evaluation of performance reports. Key among his results, the author finds an inverted-U relation between judgment accuracy and the level of information supplied to the decision maker.

Similarly, Casey (1980) examines the effect of information load on judgment accuracy. The experiment used bank loan officers as participants to make bankruptcy

judgments on ten sample firms. The loan officers were assigned to one of three information load conditions. The author found that the accuracy of the loan officers was significantly better with higher information load than the lesser load. Also, the author found that the predictive accuracy of those with the heaviest information load did not differ from those with the heavy load, but that those with the heaviest information load took significantly more time to assimilate the data. While this later finding might appear to contradict the information overload hypothesis, two observations are helpful in understanding this finding. First, while testing the potential inverted-U pattern of information overload requires a minimum of three measurement points, actually detecting this relation may require more – a limitation not noted in Casey (1980). Second, performance (i.e., predictive accuracy) of those with the highest information load may have been reduced had they been required to perform in the same amount of time taken by those with the moderate information load. This conjecture is supported by the underlying argument for the information overload hypothesis that individuals are limited in their ability to assimilate information *for a given period of time* (Jacoby et al., 1974). In support of this notion, Tuttle and Burton (1999) suggest that information overload is not just a function of total level of information, but rather a function of total information *per unit of time*. This concept relates to the present study, in that individuals given a constant amount of time to perform the task should exhibit information overload effects as the total level of information increases (i.e., with increases in feedback delay).

Kachelmeier and Granof (1993) attribute information overload to the unexpected finding that individuals provided both depreciation and a detailed historical recap of performance led to lower ratings and capital expenditures than when either of these information elements were given in isolation.

In sum, these studies demonstrate that the concept of information overload is applicable to various tasks of interest to managers and management accountants. The information overload literature both in and out of accounting share a common theme that some information is helpful for decision makers to improve the quality of their decisions and learn, but that too much information can be a detriment to decision quality and learning, particularly if the amount of time to process the information is held constant (see arguments in Tuttle and Burton, 1999).

This study differs from prior studies on information overload in two ways. First, task information load is manipulated in an indirect way through the timing of outcome-based performance feedback. Second, total information load in this study is a function of both the level of feedback delay (above) *and* individuals' choice to seek additional information.

3.3 Hypothesis Development

When performance feedback is given, individuals are confronted with the intertemporal choice to either use the feedback to learn the task better, which may improve future performance at the expense of current performance, or to use the feedback only as guidance to perform the task better now, which will improve current performance

at the expense of future performance. In this study, individuals given immediate feedback face this tradeoff at a point when the amount of task information available at the time feedback is given is lower than the minimum level of task information needed to learn the cues. As such, individuals given immediate feedback need to seek additional task information after receiving feedback in order to be able to learn the cues (see the area labeled "A" in Figure 1). This costly search for information (Gabaix et al., 2006) carries an uncertain future reward of better future performance since learning the cues in the task is dependent on many factors in addition to the choice to seek additional information (e.g., such as individual skill, mental alertness, luck, etc.). The intertemporal choice of seeking more task information favors a myopic focus on short-term performance (i.e., payoffs) because the future rewards of a current investment in obtaining more task information are uncertain (Loewenstein and Prelec, 1992; Keren and Roelofsma, 1995; Anderhub et al., 2001; Hales and Williamson, 2009). This notion is also consistent with research that finds that individuals often fail to completely consider the long-term effects of their current behavior (Thaler and Shefrin, 1981).

Given the above discussion, I expect individuals given feedback after no delay to, on average, make the intertemporal choice to not seek additional task information in the current period that could lead to high current performance at the expense of future performance (i.e., trading off increased future performance for increased current performance). Individuals given feedback after an intermediate delay are obliged to receive more task information (due to the delay of feedback) and have sufficient task

information available to learn the cues when feedback is given (see Figure 1). As such, these individuals do not need to seek additional task information in order to learn the cues, though additional task information may still be beneficial. Thus, these individuals have less of a need to seek additional task information and, as such, will not suffer performance decreases in the future period due to a failure to seek additional task information. As individuals given immediate feedback are unlikely to seek the needed additional task information, their learning of the task will suffer compared to individuals who are obligated to receive sufficient task information (through an intermediate delay of feedback) to learn the cues. This will result in a difference in learning and future performance, hypothesized below:

> H1: Future performance will be lower for individuals given outcome-based performance feedback after no delay than for individuals given feedback after intermediate delays.

While performance can arguably benefit from some delay in performance feedback, any such performance effect could eventually reverse after too much of a delay. As the length of the delay increases, two results are likely to occur. First, the salience of the intertemporal tradeoff will diminish as individuals are obligated to receive more task information in the current period (caused by a delay of performance feedback). Second, individuals' ability to learn can diminish with an increase in the delay of performance feedback (Lewis and Anderson, 1985). Individuals given outcome-based performance feedback after an extended delay have the most task information available to them at the time they receive the feedback (see the area labeled "B" in Figure 1). With

unlimited cognitive capacity, this should lead to better future performance. However, there is a point at which more information disrupts the learning process. This notion is consistent with research on information overload (Jacoby et al., 1974; Shields, 1983; Iselin, 1988; Kachelmeier and Granof, 1993), which documents negative performance effects associated with information levels beyond a certain point. This leads to the following hypothesis:

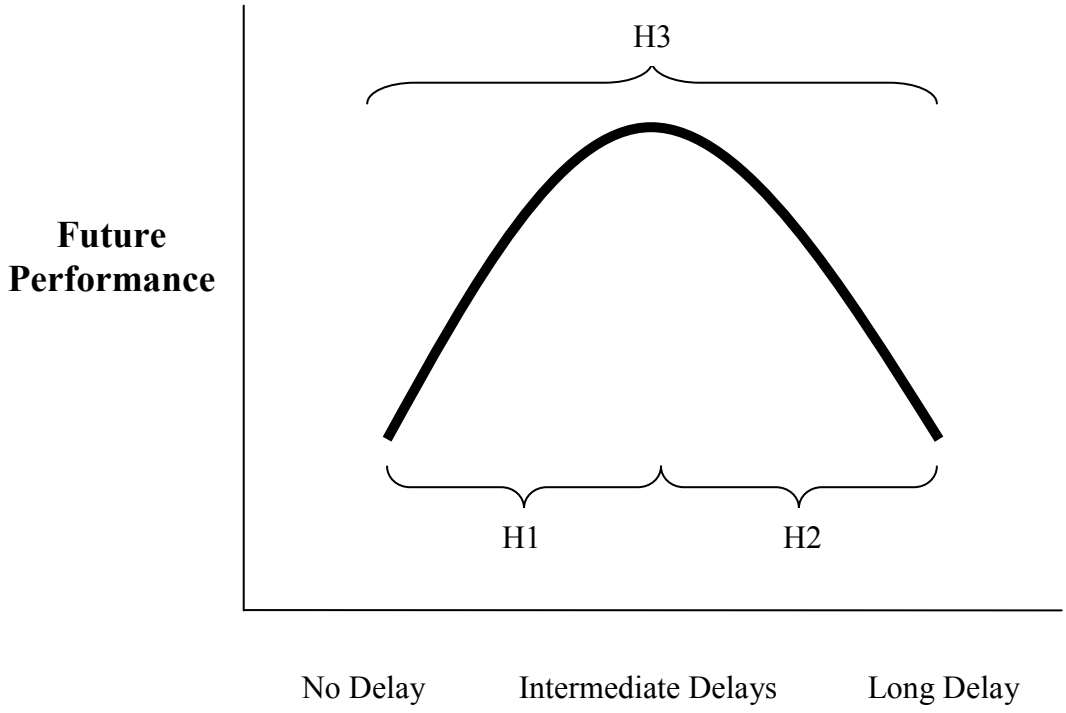
> H2: Future performance will be lower for individuals given outcome-based performance feedback after a long delay than for individuals given feedback after intermediate delays.

By combining the expectations in H1 and H2, the inverted-U relation between the timing of feedback and future performance becomes evident and is formally hypothesized below:

> H3: Future performance is an inverted-U function of the delay in performance feedback (i.e., delays in performance feedback have a non-monotonic effect on future performance).

To highlight the main tensions of the study, I predict that an increase of performance feedback delay will overcome the myopic tendency to not search for more task information by obligating the individuals to receive more information before receiving feedback, which should benefit future performance (H1). However, I also predict that delaying the provision of performance feedback too much will lead to confusion and information overload, which will lead to poorer future performance (H2). This results in an inverted-U relation between the delay of outcome-based performance feedback and future performance (H3). Figure 2 graphically presents H1, H2, and H3.

FIGURE 2
Graphical Representation of Hypotheses 1, 2, and 3



Chapter 4: Maze Experiment

4.1 Experimental Design

4.1.1 Participants

Eighty-four accounting students recruited from upper division accounting courses participated in the experiment. The participants were 22.4 years old, on average, and had 3.7 years of college experience. Fifty-one percent of the participants were female. Participants were randomly assigned to one of four experimental conditions. Participants worked on the task individually and were given monetary compensation for participating in the study.

4.1.2 Instructions and Task

The task employed to test the hypotheses is a computerized maze task adapted from Lewis and Anderson (1985). This task captures important aspects of many decision-making settings in which employees choose a path to follow and have the ability to learn from the task environment. Increases in the delay of feedback in this task require individuals to be exposed to increasing amounts of task information (e.g., in the form of seeing more of the task environment before receiving feedback), a characteristic found in tasks of many modern organizations. However, a key feature of the task is the ability for participants to seek additional task information (e.g., by searching around the maze) to better learn how to perform the task in the future, irrespective of when they receive outcome feedback. This task does not allow for much, if any, transfer of previously learned skills, which effectively controls for prior knowledge.

Participants were informed that they would be participating in two periods of the maze task, comprising twelve minutes in the first period and twenty-four minutes in the second period. They were asked to complete as many mazes as possible in each period. All participants were paid for their performance during the two periods of the experiment immediately after completing the post-experimental questionnaire. Participants received $0.50 for each maze completed up to the disclosed maximum of twenty mazes in the first period and forty mazes in the second period. Additionally, any participant who finished the allotted mazes in either period received a $0.01 per second bonus for any time remaining upon completion of the mazes.[5] The magnitude of potential second-period payout was emphasized through a quiz immediately prior to performing the task.

Each maze consists of a series of connected rooms that fan out from each other like roots and, as such, are not interconnected. This implies that there is only one correct path to the end of the maze (i.e., the "Treasure Room"). Included in the participant instructions was a schematic of a sample maze, as shown in Appendix A.

Also given to participants was a screenshot from a sample room within a maze (see Appendix A). Each room has four doors – one from which the participant enters and three from which to choose in order to proceed further in the maze.[6] In both the correct and incorrect rooms, there are visual cues (e.g., presence or absence of certain key objects) from which participants can learn that they have entered an incorrect path.

[5] This $0.01 per second bonus for finishing the maximum mazes before time expires in the period was included in the experimental design to encourage participants to place a value on their time/effort spent in the task (Sprinkle, 2000).

[6] The starting room of each maze only has three doors.

Learning the cue patterns is crucial to performance, particularly when performance feedback is unavailable. For the three types of mazes in the task, each distinguished by a different color, there is a specific object that is present on incorrect paths and absent on correct paths. Only by searching both correct and incorrect rooms can these cues be discovered. This feature of the task is key to understanding participants' intertemporal decisions. All participants were instructed that this was the way in which cues could be learned.[7] As such, decisions to not navigate through incorrect paths reveal myopic behavior. All participants were also informed of the importance of learning the cue patterns and that the cue patterns would be consistent between the first and second periods.[8] The instructions to and structure of the task emphasize to participants the need to learn the cue patterns in the first period in order to maximize total compensation across both periods.

The experiment consists of two periods. In the first period (i.e., "learning phase"), individuals are given outcome-based feedback. In the second period (i.e., "post-learning phase"), no participants are given feedback. All participants were informed of this feature of the task in the instructions. This design allows the effect that the timing of feedback has on future performance to be separated from its effect on current

[7] From the experimental instructions – "[T]here are TWO ways to navigate through a maze. The first way is to search through the maze until you find the Treasure Room. The second way is to identify visual "cues" that indicate that you are on the INCORRECT path. As such, it would be beneficial to spend time studying incorrect paths. Using these cues will assist in a more timely completion of the mazes in the first round as well as the mazes in the second round."

[8] These instructions (i.e., process feedback) were given to all participants to create a common mental representation of the task across conditions. The performance (i.e., outcome) feedback is given to trigger information search behaviors to clarify the provided mental model, not create a new one. However, in this study, I am unable to determine which participants had an accurate mental model of the task outside of their performance behaviors.

performance (Salmoni et al., 1984). This feature of the design (i.e., the removal of feedback for the second period) allows participants to demonstrate their learning of the task, which cannot be clearly observed if the feedback is always present. Further, this feature maps into many real-world settings in which performance feedback is significantly reduced or becomes unavailable.[9] Figure 3 shows a summary timeline of the experiment.

4.1.3 Experimental Design and Treatments

I test the hypotheses using a 1×4 between-subject experimental design, manipulating the delay of the outcome-based performance feedback (feedback given after no delay, after a short-intermediate delay, after a long-intermediate delay, or after a long delay) in the first period of the experimental task. In the second period, feedback is unavailable for all participants.

In the first period, participants given feedback after no delay were notified by a message that appeared immediately if a selected path was incorrect. If the selected direction was correct, the participant proceeded forward. Participants given feedback after a short-[long-]intermediate delay were notified that they had entered an incorrect path once they entered the first [second] room on the selected incorrect path. Participants given feedback after a long delay were notified that they were on an incorrect path once they entered the third room on a selected incorrect path, which is the dead-end room.

[9] For example, one's supervisor may not be readily available to give performance feedback. Another example is a situation in which, after an individual is trained to perform a task (performance feedback being readily available during training), the nature of the task requires an individual to perform for extended periods of time without performance feedback (i.e., a field service representative, etc.).

In order to keep *total* task information availability constant, all participants were free to continue down any incorrect path, regardless of feedback condition, and all were informed of the availability of this action. The quantity of task information given to individuals *at the time they first receive outcome-based performance feedback* depends on the condition to which they were assigned. For example, those assigned to the no delay condition are only forced to see the first room before receiving feedback on the accuracy of their directional choice, whereas those assigned to the short-intermediate delay condition must enter an additional room before receiving feedback on their directional choice. The instructions and comprehension checks were designed to ensure that participants understood that more task information can be gained by continuing down incorrect paths. Additionally, this strategy was highlighted as the way to best learn the cues contained in the mazes.

In the second period, all participants were no longer given the feedback message regarding their directional choices.

4.1.4 Dependent Measures

The main dependent variable of interest is performance in the post-learning phase (i.e., second period), measured as the number of mazes participants completed per minute during the respective period. Given that outcome-based feedback is not present in the post-learning phase, participants can more clearly demonstrate their learning of the task in this phase, thus isolating the differential effects of the timing of feedback from the learning phase on future performance. I similarly measure performance in the learning

phase (i.e., first period) in order to facilitate comparative performance between the two periods.

4.2 Results

4.2.1 Tests of Hypotheses

Table 1 presents the mean number of mazes completed per minute, total mazes completed, and compensation earned for both the learning phase (Panel A) and the post-learning phase (Panel B). Table 1 also presents mazes per minute ranked against all other participants in the experiment (with the best performer given the highest rank). I use the ranked performance per minute in my analyses for two reasons. First, econometric concerns have been shown to arise from non-normal distributions (Friedman, 1937; Conover and Iman, 1981), which can be mitigated by using ranks in the statistical analyses. The non-normality in this study arises from the nature of the task, in which those who learn the cues complete all of the allotted mazes, while those who do not learn the cues complete few mazes (i.e., most observations fall on the extreme left or extreme right of the performance spectrum). Second, in order to facilitate comparison of participant performance between the learning and post-learning phase, ranking performance allows for a clearer measure of a participant's performance relative to other participants in the study. This leads to a more complete view of the tradeoffs made by participants by removing task-specific characteristics of performance, which are analyzed in the after testing the hypotheses. Figure 4 graphs the primary dependent variable of the

experiment. Specifically, Figure 4 shows the mean ranked number of mazes completed per minute by each performance feedback condition for the post-learning phase.

The main dependent variable of interest for H1 and H2 is the ranked number of mazes completed per minute in the post-learning phase.[10] Table 2, Panel B reports the ANOVA of this variable for all four performance feedback conditions, showing that the difference between the four conditions is statistically significant. The two intermediate delay conditions do not statistically differ from each other ($p = 0.890$, two-sided). As such, these conditions are collapsed for the remaining comparisons.[11]

The purpose of H1 [H2] is to test whether participants given performance feedback after no delay [a long delay] perform worse than participants given performance feedback after intermediate delays. The tests for H1 and H2 are the univariate comparisons for ranked performance per minute in the post-learning phase (i.e., the second period). Table 2, Panel C presents these univariate comparisons between the performance feedback conditions. The difference between the ranked performance of participants in the no delay and intermediate delay performance feedback conditions is statistically significant ($p = 0.001$, one-sided), with participants given performance feedback after no delay completing fewer mazes than participants given performance

[10] The ex ante strategy of finishing the allotted mazes in the first period as fast as possible without regard for learning the cues patterns (with the accompanying second period benefit) is not superior to ex ante strategy of learning the cue patterns in the first period and reaping the performance rewards in the second period.

[11] In testing for any inverted-U relation, identification of the apex of the curve is a challenge. As such, I designed this experiment to be flexible to the inclusion of multiple intermediate delays. Using either intermediate delay condition separately in the univariate comparisons does not substantially alter the inferences of the findings.

feedback after an intermediate delay.[12] The difference between the ranked average performance for participants in the intermediate and long delay performance feedback conditions is marginally significant ($p = 0.059$, one-sided), with participants given performance feedback after a long delay completing fewer mazes than participants given performance feedback after an intermediate delay. These results strongly support H1 and provide marginal support for H2.

The pattern of means follows the hypothesized inverted-U relation of H3. To statistically examine this relation, I perform a one-way ANOVA trend analysis with orthogonal polynomial contrasts (Rosenthal and Rosnow, 1991; Kirk, 1995). I find that the linear trend is insignificant ($F = 2.625$, $p = 0.109$), while the quadratic trend is highly significant ($F = 10.002$, $p = 0.002$).[13] This result supports H3.

Using the unranked performance in the post-learning phase does not substantially alter the implications of this study. For example, the tests for H1 using the number of mazes completed and the number of mazes completed per minute result in significant differences [($t = 3.446$; $p < 0.001$, one-sided) and ($t = 3.107$; $p = 0.002$, one-sided), respectively]. For test for H2 using these measures results in weaker but still moderately significant differences [($t = 1.030$; $p = 0.153$, one-sided) and ($t = 1.475$; $p = 0.072$, one-sided), respectively]. To test H3 with these dependent measures for post-learning phase

[12] Some might note that this finding may be due to the concept of outcome-irrelevant learning structures (OILS) (Einhorn, 1980; Einhorn and Hogarth, 1981), which refers to the idea that rapid [positive] outcome feedback can be irrelevant or even harmful to learning when knowledge of the task structure is lacking or in error. Einhorn notes that this concept is relevant to decision settings in which there are few instances of outcome feedback. In my setting, participants face *many* instances of negative outcome feedback, allowing them to gain a more complete understanding of the task.

[13] The cubic trend is insignificant ($F = 0.151$, $p = 0.699$).

performance results in a significant quadratic trend [($F = 7.517$, $p = 0.008$) and ($F = 9.091$, $p = 0.003$), respectively] and an insignificant linear trend [($F = 2.614$, $p = 0.110$) and ($F = 1.288$, $p = 0.260$), respectively].[14]

4.2.2 Supplemental Analyses

The purposes of these supplemental analyses are 1) to examine alternative measures of performance to better understand the intertemporal tradeoffs that participants faced in this research setting, 2) to better understand the processes underlying the differences predicted in H1 by examining participants' actual behavior and decisions, and 3) to gain confidence that the information environment within the task supports the underlying information overload theory of H2.

4.2.2.1 Alternative Performance Measures

Since participants were compensated for performance in both periods, examining the effect of feedback timing on *total* compensation and *total* performance would be helpful to understanding participant behavior over both periods in the task. Table 3 [Table 4] presents the ANOVA and univariate comparisons of the effect of feedback timing on total compensation [total performance as measured by mazes per minute].[15] Of particular note in Table 3 [Table 4], Panel C, the univariate comparisons testing H1 and H2 are significant (all with $p < 0.05$, one-sided). Also, the trend analysis for total compensation [total performance] results in a significant quadratic trend ($F = 6.016$, $p =$

[14] The cubic trends are insignificant [($F = 1.162$, $p = 0.284$) and ($F = 0.735$, $p = 0.394$), respectively].
[15] By dividing individual performance (i.e., mazes completed) by the amount of time spent in the mazes, I can account for the speed of performance, which biases against finding results consistent with my hypotheses by amplifying the performance of participants who myopically focus on short-term performance.

0.001 [$F = 11.001$, $p = 0.001$]) and an insignificant linear trend ($F = 0.227$, $p = 0.635$ [$F = 1.509$, $p = 0.223$]), in support of H3. [16] In sum, employing total compensation and total performance as dependent measures of individual performance to test hypotheses H1, H2, and H3 does not change the inferences of the main findings of this study.

Another potentially relevant measure of performance is participants' comparative performance between the first and second period as this can shed light on how they responded to the intertemporal tradeoffs introduced in this setting. Table 5, Panel A presents summary data on the mean performance rank for participants in the first period (i.e., learning phase). Participants with less of a delay of performance feedback completed significantly ($p < 0.001$, two-sided) more mazes per minute during the learning phase, supported by the ANOVA results in Table 5, Panel B. The use of ranked performance provides a robust measure of the tradeoffs that participants chose allowing us to compare each participant's standing (i.e., rank) in each period with that of all other participants in the study. Figure 5 graphically presents the ranked performance for the learning and post-learning phases by delay condition and highlights the tension of the intertemporal tradeoff facing participants in this setting. Since a delay in feedback reduces the salience of the intertemporal tradeoff by forcing individuals with more of a delay to receive more task information, I restrict the statistical analysis of the difference in performance to individuals given feedback after no and short-intermediate delays. Participants in the no delay feedback condition experienced a 37.17 rank decrease

[16] The cubic trend is also insignificant ($F = 0.606$, $p = 0.438$).

between the first and second periods, indicating a focus on high performance in the first period but with little learning of the task to benefit performance in the second period. Participants in the short-intermediate delay condition experienced a 0.19 rank increase between the first and second periods. The change from the first to second period of performance rank is significantly different between participants in the no-delay condition and those in the short-intermediate delay condition ($t = 3.640$; $p = 0.001$, two-sided). These findings support the notion that individuals given immediate feedback use the feedback as a crutch to facilitate current performance at the expense of future performance.

4.2.2.2 Participant Behavior within Task

To investigate how the outcome-based performance feedback affected participants' behavior, I isolate all of the movement decisions made immediately following the receipt of the feedback in order to determine the extent to which participants sought additional, beneficial task information after receiving the feedback. This search for additional information represents a costly decision for the participants, as they were compensated for the number of mazes completed *and* for the time remaining after completing the allotted number of mazes. It is also important to note that all participants were informed of the benefit to future performance of learning the cues and that the information for learning was contained in both correct *and* incorrect paths of the mazes. All participants confirmed their knowledge of the learning of cues in the experimental materials. By investigating the decisions of individuals immediately

following the initial receipt of the performance feedback, one can make inferences on the individuals' use of the information contained in the feedback. For example, if an individual receives the feedback message that s/he chose an incorrect path, the individual can choose to either seek additional beneficial information down the incorrect path, which can benefit future performance, or can choose to *not* seek additional information down the incorrect path by choosing a different path, which benefits short-term performance. Presumably, by choosing to seek additional information down the incorrect path, the individual is attempting to learn the cues, as to do otherwise would decrease participant payoff. Across all conditions, I find that participants proceeded down known incorrect paths (to seek additional beneficial information) in only 7% of the available instances. This supports the notion that individuals tend to behave myopically when trading off current and future benefits of one's actions.

To investigate whether participants' information search behavior (after receipt of performance feedback) differs due to the timing of feedback, I compare the behavior of participants given immediate feedback with that of those given feedback after intermediate delays. First, I examine the extent to which participants never proceeded down known incorrect paths. Of the twenty-one participants in each of the no, short-intermediate, and long-intermediate delay conditions, eleven, two, and three participants, respectively, never sought additional task information. These relative frequencies are significantly different ($\chi^2 = 12.231$; $p = .002$). This supports the notion that information search behavior differs between levels of delay of performance feedback. Second, I

compare the ranked extent to which participants *did* proceed down known incorrect paths.[17] Table 6 presents the ANOVA and univariate comparison of information search behavior between the feedback groups during the learning phase (i.e., the first period). Participants given immediate feedback sought additional information significantly less often than participants given performance feedback after intermediate delays ($p = 0.002$, two-sided). This difference is greater when examining behavior early in the first period, when most of the first period learning took place ($p = 0.001$, two-sided, untabulated). In sum, participants with the least amount of task information when the feedback is given and who would benefit most from searching for additional task information (i.e., those in the immediate feedback condition) engaged in additional task information search less often than participants given more task information (i.e., those in the intermediate-delay conditions). This suggests that the timing of outcome-based performance feedback increases the level of myopia exhibited by individuals given immediate feedback.

The following are some representative comments left by two participants in the no-delay feedback condition that illustrate the tension faced by participants in this condition (and possible explanation for the differential information search patterns), the tenor of which was not present in any other feedback condition. "I felt like the money was so easy to make in the first round, I hardly spent any time thinking about the possibility of making more cash in the second round. I feel like this was a horrible decision." "I wanted to find cue patterns in the first round, but the ease at which I was

[17] The ranked extent of proceeding down known incorrect paths is used to again address the econometric concern of non-normality, which arises because of the large amount of participants (16 out of 63) who never proceeded down known incorrect paths to seek additional task information.

able to get to the finish by randomly clicking was too attractive to waste much time." This anecdotal evidence from the comments of participants suggests that feedback given after no delay served to amplify the participants' myopic behavior following receipt of feedback.

4.2.2.3 Information Environment within Task

The level of task information available to the individual is central to the hypotheses of this study, particularly H2 which relies on theory from the information overload literature. The purpose of this analysis is to present various measures of information load to ensure that the manipulation of the delay of feedback did, in fact, also manipulate task information level.

To check the manipulation of total level of task information available to the participants during the first period, I look at four proxies for this construct: 1) total number of rooms entered per maze, 2) total time spent per maze, 3) mix of incorrect and correct rooms entered per maze, and 4) percent of time spent on incorrect paths. The first two proxies assume that all rooms and time spent in the mazes provide an equivalent incremental amount of task information. The second two proxies separate the information value of rooms of correct and incorrect paths, as individuals need information on both correct *and* incorrect paths to learn the cues in the mazes. Table 7 presents descriptive statistics for each of these task information constructs. Participants with less of a delay of outcome-based performance feedback entered less rooms per maze, spent less time on each maze, entered more correct rooms compared to incorrect

rooms, and spent more time in correct paths than on incorrect paths. In untabulated univariate comparisons, the task information differences between each condition are significant (with one exception between the short- and long-intermediate delay conditions for total time spent per maze).

This analysis corroborates the reasoning that increasing outcome-based feedback delay increased the total level of task information available to participants. Given that participants provided with feedback after a long delay were exposed to the highest information load, the information overload hypothesis for the performance difference between these participants and participants given feedback after intermediate delays appears to be appropriate in this study.

FIGURE 3
Summary Timeline of Experiment

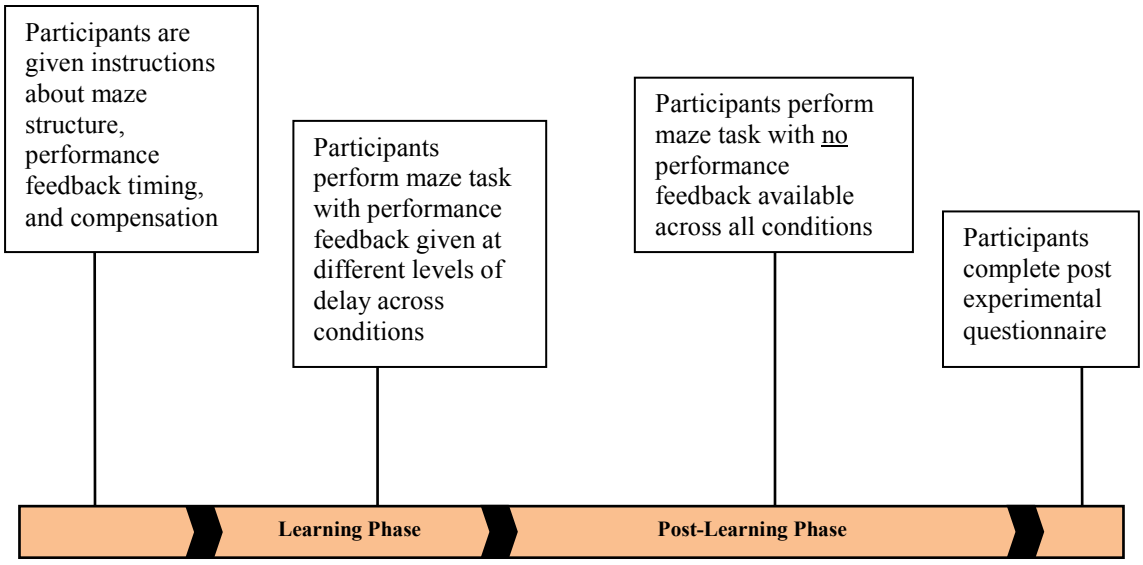


FIGURE 4
Mean Ranked Number of Mazes Completed per Minute in Post-Learning Phase by Feedback Delay

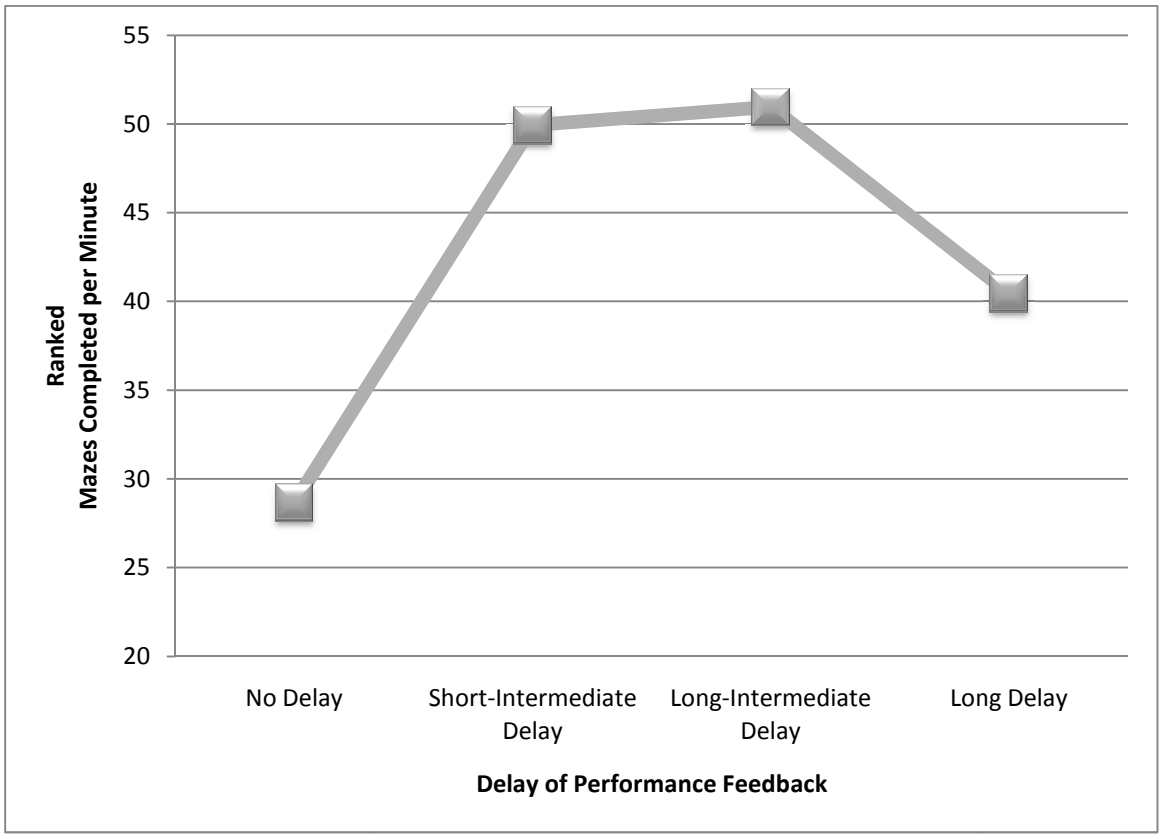


FIGURE 5
Mean Ranked Number of Mazes Completed per Minute by Feedback Delay

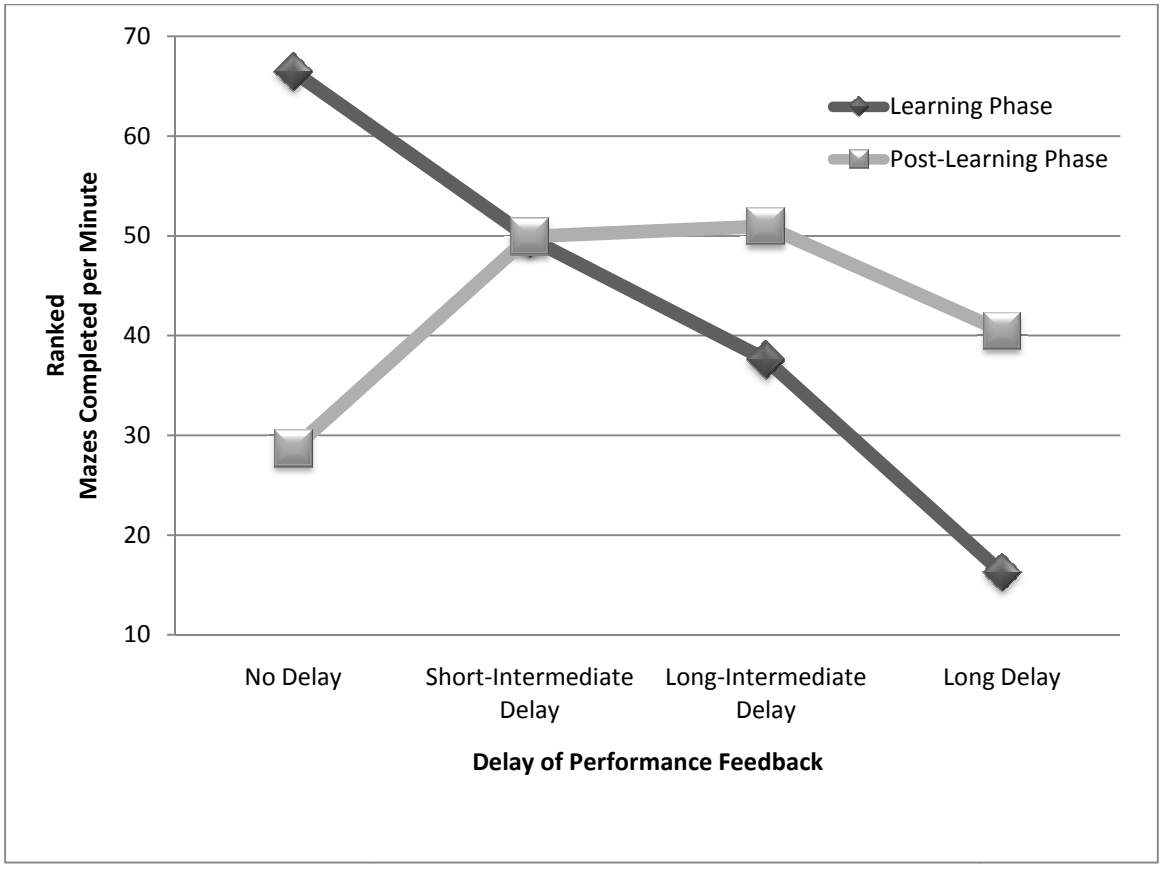


TABLE 1
Descriptive Statistics

Panel A: Learning Phase Performance

Timing of Performance Feedback in Learning Phase	Mean Mazes Completed per Minute (Standard Deviation)	Mean Mazes Completed (Standard Deviation)	Compensation (in $)	Mean Rank of Mazes Completed per Minute (Standard Deviation)	Number of Subjects
No Delay	3.24 (1.047)	19.67 (1.528)	13.02	66.45 (14.625)	21
Short-Intermediate Delay	2.33 (1.046)	18.05 (4.092)	10.90	49.74 (22.206)	21
Long-Intermediate Delay	1.70 (0.562)	17.48 (4.020)	9.51	37.57 (14.492)	21
Long Delay	0.84 (0.472)	9.81 (4.996)	4.98	16.24 (11.995)	21

Panel B: Post-Learning Phase Performance

Timing of Performance Feedback in Learning Phase	Mean Mazes Completed per Minute (Standard Deviation)	Mean Mazes Completed (Standard Deviation)	Compensation (in $)	Mean Rank of Mazes Completed per Minute (Standard Deviation)	Number of Subjects
No Delay	0.19 (0.248)	4.52 (1.528)	2.26	28.67 (20.921)	21
Short-Intermediate Delay	1.49 (1.657)	18.52 (17.733)	11.71	49.93 (24.284)	21
Long-Intermediate Delay	1.22 (1.659)	16.24 (15.543)	9.97	50.95 (23.387)	21
Long Delay	0.73 (1.081)	12.90 (15.741)	7.34	40.45 (23.316)	21

TABLE 2
ANOVA and Univariate Comparisons for the Effect of Timing of Performance Feedback on Ranked Completed Mazes per Minute in Post-Learning Phase

Panel A: Descriptive Statistics

Timing of Performance Feedback in Learning Phase	**Mean Rank**	**Standard Deviation**	**Number of Subjects**
No Delay	28.67	20.921	21
Short-Intermediate Delay	49.93	24.284	21
Long-Intermediate Delay	50.95	23.387	21
Long Delay	40.45	23.316	21

Panel B: Analysis of Variance

Factor	*df*	*Mean Square*	*F*	*p-value*[a]
TIMING OF PERFORMANCE FEEDBACK	3	2255.262	4.259	0.008
ERROR	80	529.503		

Panel C: Univariate Comparisons

	Univariate Statistics – t-statistic (p-value[a])	
	No Delay versus Intermediate Delays	*Intermediate Delays versus Long Delay*
RANKED MAZES COMPLETED PER MINUTE IN POST-LEARNING PHASE	3.586 **(0.001)**	1.592 **(0.059)**

[a] Reported p-values are two-tailed unless testing a one-tailed prediction as indicated by bold face.

TABLE 3
ANOVA and Univariate Comparisons for the Effect of Timing of Performance Feedback on Total Compensation

Panel A: Descriptive Statistics

Timing of Performance Feedback in Learning Phase	**Mean (in $)**	**Standard Deviation**	**Number of Subjects**
No Delay	15.28	2.987	21
Short-Intermediate Delay	22.61	12.140	21
Long-Intermediate Delay	19.48	11.725	21
Long Delay	12.32	10.359	21

Panel B: Analysis of Variance

Factor	*df*	*Mean Square*	*F*	*p-value*[a]
TIMING OF PERFORMANCE FEEDBACK	3	432.622	4.314	0.007
ERROR	80	100.275		

Panel C: Univariate Comparisons

	Univariate Statistics – t-statistic (p-value[a])	
	No Delay versus Intermediate Delays	*Intermediate Delays versus Long Delay*
TOTAL COMPENSATION (IN $)	2.180 **(0.017)**	2.861 **(0.003)**

[a] Reported p-values are two-tailed unless testing a one-tailed prediction as indicated by bold face.

TABLE 4
ANOVA and Univariate Comparisons for the Effect of Timing of Performance Feedback on Total Performance (Mazes per Minute)

Panel A: Descriptive Statistics

	Mean	Standard Deviation	Number of Subjects
Timing of Performance Feedback in Learning Phase			
No Delay	0.79	0.179	21
Short-Intermediate Delay	1.48	1.071	21
Long-Intermediate Delay	1.26	0.999	21
Long Delay	0.74	0.721	21

Panel B: Analysis of Variance

Factor	*df*	*Mean Square*	*F*	*p-value*[a]
TIMING OF PERFORMANCE FEEDBACK	3	2.756	4.086	0.009
ERROR	80	0.674		

Panel C: Univariate Comparisons

	Univariate Statistics – t-statistic (p-value[a])	
	No Delay versus Intermediate Delays	*Intermediate Delays versus Long Delay*
TOTAL MAZES COMPLETED PER MINUTE	2.551 **(0.01)**	2.516 **(0.01)**

[a] Reported p-values are two-tailed unless testing a one-tailed prediction as indicated by bold face.

TABLE 5
ANOVA and Univariate Comparisons for the Effect of Timing of Performance Feedback on Ranked Completed Mazes per Minute in Learning Phase

Panel A: Descriptive Statistics

Timing of Performance Feedback in Learning Phase	**Mean Rank**	**Standard Deviation**	**Number of Subjects**
No Delay	66.45	14.625	21
Short-Intermediate Delay	49.74	22.206	21
Long-Intermediate Delay	37.57	14.492	21
Long Delay	16.24	11.995	21

Panel B: Analysis of Variance

Factor	*df*	*Mean Square*	*F*	*p-value*[a]
TIMING OF PERFORMANCE FEEDBACK	3	9380.595	35.368	<0.001
ERROR	80	235.228		

[a] Reported p-values are two-tailed unless testing a one-tailed prediction as indicated by bold face.

TABLE 6
ANOVA and Univariate Comparisons for the Effect of Timing of Performance Feedback on Ranked Extent of Beneficial Information Search

Panel A: Descriptive Statistics

Timing of Performance Feedback in Learning Phase[a]	**Mean Rank**	**Standard Deviation**	**Number of Subjects**
No Delay	22.14	17.953	21
Short-Intermediate Delay	37.02	15.718	21
Long-Intermediate Delay	36.83	17.380	21

Panel B: Analysis of Variance

Factor	*df*	*Mean Square*	*F*	*p-value*[b]
TIMING OF PERFORMANCE FEEDBACK	2	1530.512	5.269	0.008
ERROR	60	290.483		

Panel C: Univariate Comparison

	Univariate Statistics – t-statistic (p-value[b]*)*
	No Delay versus Intermediate Delays
RANKED EXTENT OF BENEFICIAL INFORMATION SEARCH	3.273 (0.002)

[a] The structure of the task does not allow participants in the long-delayed performance feedback to seek additional information down the incorrect path as all must proceed backwards out of the dead end.
[b] Reported p-values are two-tailed unless testing a one-tailed prediction as indicated by bold face.

TABLE 7
Descriptive Statistics for Proxies of First Period Total Level of Task Information

	Mean Rooms Entered Per Maze (Standard Deviation)	Mean Seconds Per Maze (Standard Deviation)	Ratio of Incorrect to Correct Rooms Entered (Standard Deviation)	Percent of Total Time Spent in Incorrect Paths (Standard Deviation)	Number of Subjects
Timing of Performance Feedback in Learning Phase					
No Delay	3.37 (0.737)	18.41 (8.206)	0.12 (0.246)	0.06 (0.094)	21
Short-Intermediate Delay	6.58 (0.664)	27.96 (21.559)	1.20 (0.237)	0.34 (0.124)	21
Long-Intermediate Delay	9.58 (1.328)	31.85 (15.581)	2.23 (0.461)	0.53 (0.084)	21
Long Delay	16.81 (5.970)	81.22 (64.253)	4.86 (2.353)	0.68 (0.095)	21

Chapter 5: Conclusion

Performance feedback is a key element of performance measurement and evaluation and is an important tool used by management and management accountants to improve employee performance and learning. The objective of this research is to examine the effect of the timing of performance feedback on individual performance in a setting in which performance feedback is available for a certain period and unavailable for a future period, a setting of particular interest to managers and management accountants. I hypothesize an inverted-U relation between the delay of performance feedback and future performance, in which future performance is lowest when performance feedback is given after no delay, is increasing in the delay of performance feedback to a point, and is decreasing in further delay of performance feedback. Further, I investigate the role that two process variables – intertemporal choice and information overload – have on the relation between the timing of outcome-based performance feedback and individual future performance.

Results support an inverted-U relation between the delay of performance feedback and future performance. Specifically, I find that performance feedback given after no delay adversely affects future performance relative to when performance feedback is given after an intermediate delay. I also find support for an extended delay in performance feedback damaging future performance relative to an intermediate delay. These results are consistent with immediate feedback interfering with the learning process by encouraging myopic behavior of focusing on current performance at the

expense of future performance, while feedback given after too long of a delay can lead to information overload, causing confusion and lowering future performance.

These results can inform management and management accountants of the potential benefits and pitfalls associated with the timing of when performance feedback is disseminated to individuals. In particular, managers should consider the myopic nature of individuals to not seek potentially necessary information needed to learn after outcome-based feedback is provided. As the timing of outcome-based performance feedback is often a characteristic that can be controlled by management, managers should bear in mind the cognitive limits of individuals to process information concerning a task, particularly given an extended delay in providing feedback.

5.1 Limitations and areas for future research

The limitations of this study highlight the need for future research on the effect that the timing of performance feedback has on future performance. Specifically, depending on the experimental condition, participants were forced to wait to receive performance feedback. However, as individuals can self-select into organizations in which performance feedback is given after certain levels of delay, future research could focus on making the timing of performance feedback an endogenous choice. Also, this study examines the effect of the timing of performance feedback under only performance-based compensation. While this type of compensation contract is relevant for many settings, examining the effect of the timing of feedback under other compensation schemes could provide interesting insight into this choice variable (i.e., the

timing of feedback) of managers and management accountants. Additionally, recent research in systems-based thinking in auditing (Peecher et al., 2007; Brewster, 2011) identifies a potential person variable (i.e., systems thinking versus reductionist thinking) that could interact with the effect of the timing of feedback and future performance and provide some insight into why some participants learned and some did not. For example, it is plausible that systems-based thinkers would not need to be forced to collect additional task information through a delay of outcome-based feedback to be able to learn the task, due to their increased ability to connect elements of a complex task, while reductionist thinkers may need this delay to learn more effectively.

Appendix A: Experimental Materials

A.1 Experimental Instructions

GENERAL INFORMATION

In this experiment, you will work independently on a task to be described in these instructions. The experiment consists of two periods and will conclude with some final questions. The entire experiment will last approximately 60 minutes. Before describing the experiment, it is important to establish two ground rules.

1. NO TALKING WITHIN OR BETWEEN SESSIONS

While I hope that you find this experiment to be enjoyable, it is also serious research. Please help me maintain control over the experiment by refraining from comments or other communication with your fellow participants in this session or with other students who might be participating in future sessions. You will be working individually during this experiment, so there is no need to communicate with other participants. If you have any questions during the experiment, please raise your hand and the administrator will assist you.

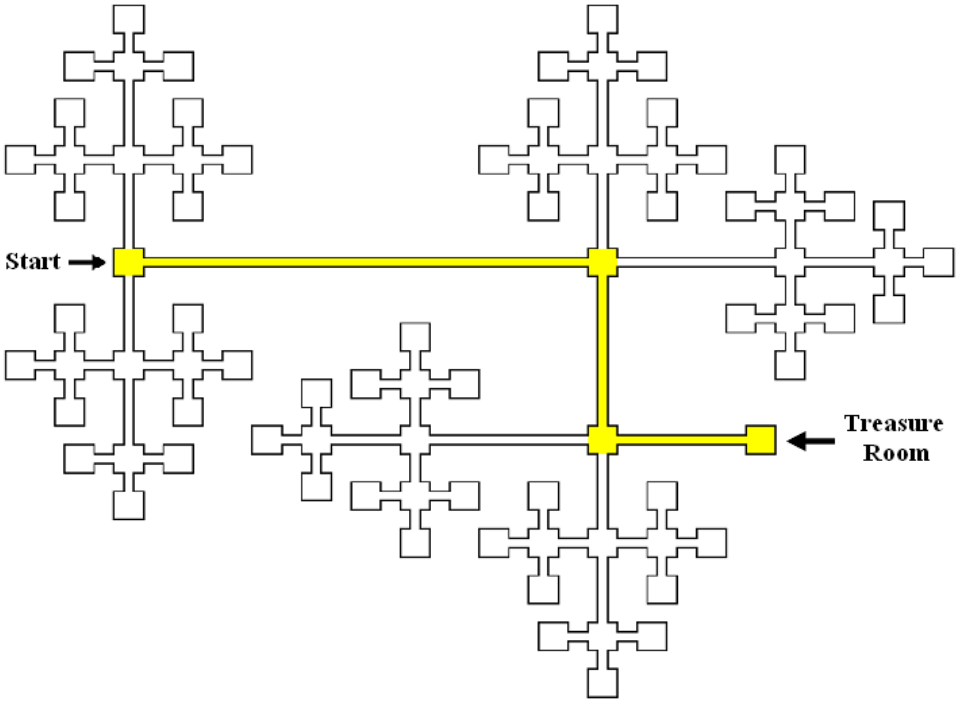
2. NO DECEPTION

The experiment will be carried out exactly as described in these instructions, with no deception of any form. Your compensation will be determined exactly as described in the rules explained later for this session.

TASK

You are asked to work on this maze task in each of two rounds. In each round, there is a series of mazes to complete. To complete each maze, you must successfully reach the Treasure Room.

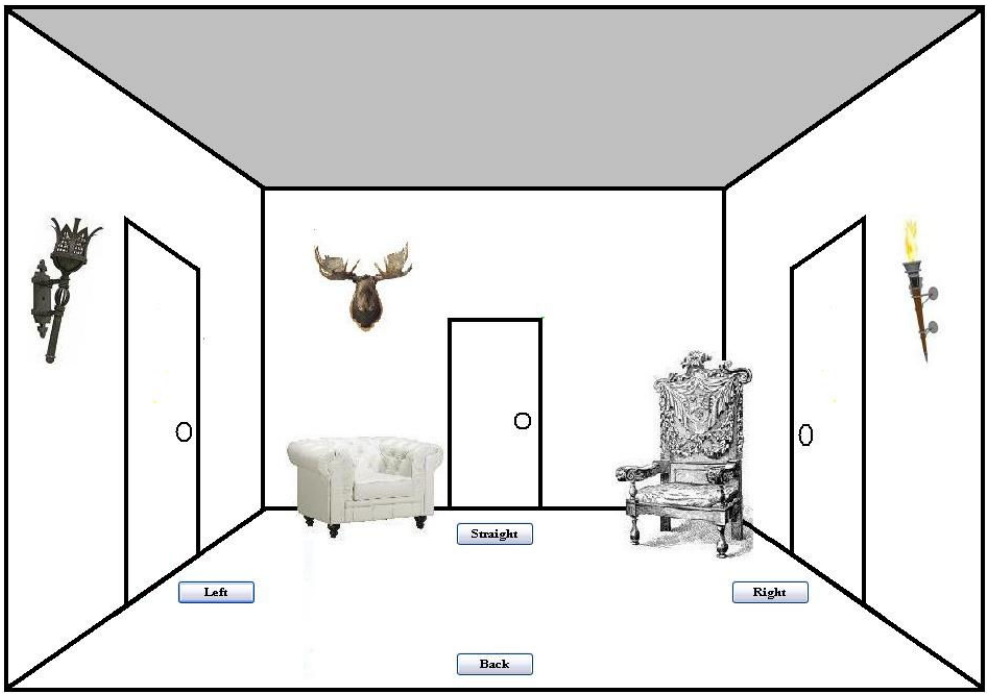
Below is a schematic of a sample maze similar to those you will work on in this task. Take a minute to familiarize yourself with the structure of the maze before continuing.



The mazes in this task consist of a series of connected rooms that fan out from each other like roots and are, therefore, NOT interconnected. This implies that there is only one correct path (noted in yellow in the example above) to the Treasure Room and that proceeding down an incorrect path will not connect you to the correct path. Also note that all dead ends are equally removed from the correct path.

TASK (Continued)

In the task program, there is a button corresponding to each directional command – Left, Straight, Right, and Back (with the exception of the starting room, which has no Back button, and dead end rooms, which have no Left, Straight, or Right buttons). You will navigate through the maze by clicking the buttons under each door. It is important to note that you are always facing the same direction in each room of the maze. To illustrate, by clicking the Back button you navigate to the room immediately preceding the room you are in and will be facing the same direction that you were facing when you had first entered that room. In other words, it is as if you just stepped backwards to the previous room (as opposed to turning around and entering the previous room). A screenshot of a typical room is given below.



Before you continue, please note that there are TWO ways to navigate through a maze. The first way is to search through the maze until you find the Treasure Room. The second way is to identify visual "cues" that indicate that you are on the INCORRECT path. As such, it would be beneficial to spend time studying incorrect paths. Using these cues will assist in a more timely completion of the mazes in the first round as well as the mazes in the second round. These cues are easier to detect in the first round than in the second round.

Throughout both rounds of this task, there are three types of mazes. Each maze type is identified by the color of doors in the maze – red, yellow, and blue. Each maze type has a different cue assigned to it. Cues consist of the presence or absence of a certain object or objects in the rooms of the maze. Cues are NOT related to the positioning of objects in each room, but only consist of the presence or absence of a certain object or objects in the rooms.

The SAME cues are used on EACH type of maze (indicated by the color of doors). So, for example, if you identify the cue for a red-doored maze, that SAME cue will apply to ALL red-doored mazes you encounter in BOTH the first and the second rounds. Thus, once you discover a cue (e.g., for the red-doored mazes), you will be able to use it repeatedly for each maze of that type throughout the first and second rounds.

You are free to choose either strategy for navigating to the Treasure Room in each round. While searching the maze is a reliable way to reach the Treasure Room, it will take more time than using the cues once they have been identified. Cues, on the other hand, will initially take time to discover but will allow you to complete the mazes much faster. Feel free to use the paper and pen at your station to keep notes.

To test your understanding of the cues please answer the following question.

Comprehension Check Questions

1. The same cues will be used in the same maze types (e.g. red, yellow, or blue) for both the first and second rounds. (True)
2. Cues indicate that you are on correct paths. (False)
3. Cues are related to the positioning of objects contained in each room. (False)

TASK (Continued)

In the FIRST round, you will be given the same type of maze (e.g. red, yellow, or blue) for three consecutive mazes followed by the second type of maze for the next three mazes followed by the third type of maze for the next three mazes. After these nine mazes, you will be given a different maze type (alternating between the three - red, yellow, and blue) for each maze until the time runs out.

In the SECOND round, you will be given the choice of which type of maze (e.g. red, yellow, or blue) to complete – up to a predetermined number of mazes for each type. At the end of each maze, you will be given the same choice unless you have previously completed all of the allotted mazes of a particular type (i.e. color), in which case you will be allowed to choose from mazes of the remaining type(s). For example, if you complete all of the allotted yellow-doored mazes, you would then only be allowed to choose a red- or a blue-doored maze. After completing all of the allotted mazes for each type (i.e. color), each subsequent maze will be of a randomly determined maze type (between the three available).

You will work on the maze task for 12 minutes in the first round and 24 minutes in the second round. There is a maximum of 20 mazes to complete in the first round and 40 mazes to complete in the second round.

FEEDBACK

No Delay

During the first round, you will immediately be notified whether or not a selected path is incorrect. An example of the notification message is seen below. However, you may proceed down incorrect paths for as long as you would like. During the second round, you will NOT be notified that a selected path is incorrect.

Short-Intermediate Delay

During the first round, you will immediately be notified once you have entered an incorrect path. An example of the notification message is seen below. However, you may continue down incorrect paths for as long as you would like. During the second round, you will NOT be notified that you have entered an incorrect path.

Long-Intermediate Delay

During the first round, you will be notified that you have entered an incorrect path after a one-room delay. An example of the notification message is seen below. However, you may continue down incorrect paths for as long as you would like. During the second round, you will NOT be notified that you have entered an incorrect path.

Long Delay

During the first round, you will be notified that you have entered an incorrect path after a two-room delay (i.e. at dead ends). An example of the notification message is seen below. However, you may continue down incorrect paths for as long as you would like. During the second round, you will NOT be notified that you have entered an incorrect path.

Comprehension Check Questions

1. During the first round, you will be notified immediately/after a one-room delay/after a two-room delay (i.e. at dead ends) that you have entered an incorrect path [whether or not a selected path is incorrect]. (True)
2. During the second round, you will be notified [whether or not a selected path is incorrect] that you have entered an incorrect path. (False)

COMPENSATION

Performance-Based

The number of mazes you complete is highly valued, particularly during the second round. Every participant's anonymous performance in the second round will be posted following the session.

Your compensation will be based on HOW MANY mazes you complete (i.e. by reaching each 'Treasure Room') in both the first and second rounds. You will be paid $0.50 in both the first and second rounds for each maze that you complete in the given time (up to a maximum of 20 in the first round and 40 in the second round). The more mazes you complete, the more money you will make. Note that the time given to complete mazes in the second round is twice the time given to complete mazes in the first round. All participants in this session will receive $0.50 for each maze completed in the allotted time for each round in cash at the end of the experiment.

In the event that you complete all of the allotted mazes for a given round, you will be paid a bonus of $0.01 per second remaining for that round. For example, if you complete all of the mazes in the first round with 1 minute remaining you will receive a bonus of $0.60 (60 seconds * $0.01).

Fixed

The number of mazes you complete is highly valued, particularly during the second round. Every participant's anonymous performance in the second round will be posted following the session.

You will receive a fixed payment for working on the maze task in each round, regardless of the number of mazes you complete (up to a maximum of 20 in the first round and 40 in the second round). All participants in this session will receive a fixed payment of $7 for participating in the first round and $14 for participating in the second round in cash at the conclusion of this experiment. Note that the time given to complete mazes in the second round is twice the time given to complete mazes in the first round.

In the event that you complete all of the allotted mazes for a given round, you will be paid a bonus of $0.01 per second remaining for that round. For example, if you complete all of the mazes in the first round with 1 minute remaining you will receive a bonus of $0.60 (60 seconds * $0.01).

Comprehension Check Questions

1. There is a maximum number of mazes you can complete in any given round. (True)
2. You will receive a fixed bonus for completing all of the allotted mazes in a given round. (False)
3. If you completed one maze in the first round and two mazes in the second round (with no time remaining in each round), what would be your total compensation? (performance-based - $1.5 /fixed - $7)
4. If you completed twenty mazes in the first round and forty mazes in the second round (with no time remaining in each round), what would be your total compensation? (performance-based - $30/fixed - $21)

INTERPERIOD

Performance-Based

In the second round, you will have 24 minutes to work on the maze task. In this round, you will NOT be notified that you have entered an incorrect path.

Please remember that the same cues you discovered in the first round can be used repeatedly for the mazes in the second round.

The number of mazes you complete is highly valued. Every participant's anonymous performance in this round will be posted following the session.

You will be paid $0.50 for each maze you complete in this round.

Please click on the button below that matches the maze type you would like to work on first (this will initiate the second round).

Fixed

In the second round, you will have 24 minutes to work on the maze task. In this round, you will NOT be notified that you have entered an incorrect path.

Please remember that the same cues you discovered in the first round can be used repeatedly for the mazes in the second round.

The number of mazes you complete is highly valued. Every participant's anonymous performance in this round will be posted following the session.

You will be paid $14 for working on the maze task in this round.

Please click on the button below that matches the maze type you would like to work on first (this will initiate the second round).

A.2 Experimental questionnaires

A.2.1 Task attractiveness questionnaire

How would you describe THE TASK?

Extremely	*Quite*	*Slightly*	*Neither one nor the other*	*Slightly*	*Quite*	*Extremely*
Attractive						**Repulsive**
1	2	3	4	5	6	7
Dull						**Exciting**
1	2	3	4	5	6	7
Good						**Bad**
1	2	3	4	5	6	7
Boring						**Interesting**
1	2	3	4	5	6	7
Superior						**Inferior**
1	2	3	4	5	6	7
Unwholesome						**Wholesome**
1	2	3	4	5	6	7
Fun						**Tedious**
1	2	3	4	5	6	7

A.2.2 Post-experimental questionnaire

Indicate your level of agreement with each of the next five items by clicking on the radio button on the scale wherever you feel appropriate.

1. I worked hard on the task.

Strongly Disagree				Neither agree nor disagree				Strongly agree
-4	-3	-2	-1	0	1	2	3	4

2. The feedback given in the first round was helpful to my performance.

Strongly Disagree				Neither agree nor disagree				Strongly agree
-4	-3	-2	-1	0	1	2	3	4

3. I felt obligated to learn the cue patterns while performing in the first round.

Strongly Disagree				Neither agree nor disagree				Strongly agree
-4	-3	-2	-1	0	1	2	3	4

4. Finding cue patterns is a type of task that I am not particularly good at performing.

Strongly Disagree				Neither agree nor disagree				Strongly agree
-4	-3	-2	-1	0	1	2	3	4

5. I did not have enough information to be able to learn the cue patterns used in this task.

Strongly Disagree				Neither agree nor disagree				Strongly agree
-4	-3	-2	-1	0	1	2	3	4

6. Approximately what percentage of your time did you spend in each round searching for cue patterns?

Round	Percentage of Time Spent Searching for Cue Patterns	
1		%
2		%

7. Please provide any additional information that you believe would be helpful to us when examining the results of this study.

Appendix B: Visual Basic Code for Maze Task

B.1 ID.vb

```
Public Module Startup
    Dim colFormCount As New Collection()

    "To count the forms as and when they are loaded.
    Sub CountForm(ByVal frm As Form)
        colFormCount.Add(frm)
    End Sub

    "To unload all the forms from the memory.
    Sub UnloadAllForms()
        Dim frmLoaded As Object
        For Each frmLoaded In colFormCount
            frmLoaded.close()
        Next
    End Sub

End Module

Public Class ID
    Public ID As Integer
    Public Feedback As Integer
    Public Compensation As Integer
    Public Rate As Single = 0.5
    Public SecondRate As Single = 0.01
    Public Fixed As Single = 7
    Public Min1 As Integer = 12
    Public Min2 As Integer = 24
    Public Max1 As Integer = 20
    Public Max2 As Integer = 40

    Public PaidTime1 As TimeSpan
    Public PaidTime2 As TimeSpan
    Public Minutes1 As Integer
    Public Minutes2 As Integer
    Public Seconds1 As Integer
    Public Seconds2 As Integer
    Public TotalSeconds1 As Integer
    Public TotalSeconds2 As Integer

    Public FinishedEarly1 As Boolean = False
    Public FinishedEarly2 As Boolean = False

    Public IntMillisecond As Integer

    'This variable is to denote which cue pattern to use
    Public Cue As Integer = 1

    'Maze colors for each cue pattern (1 through 3)
    Public Color1 As Integer
    Public Color2 As Integer
    Public Color3 As Integer

    'Counter for the quotas of maze colors for the second round
    Public Quota As Integer = 5
    Public Counter1Red As Integer
    Public Counter2Blue As Integer
    Public Counter3Yellow As Integer
```

```
'This code is for keeping track of the number of mazes and the period - needs to be in an open form, likely the one with the timer
Public Maze As Integer = 1
Public Period As Integer = 1
Public CompletedMaze1 As Integer
Public CompletedMaze2 As Integer
Public MazeComp1 As Single
Public MazeComp2 As Single
Public SecComp1 As Single
Public SecComp2 As Single
Public CashComp1 As String
Public CashComp2 As String
Public Comp1 As Single
Public Comp2 As Single
Public AlarmTime As Date
Public RemainingTime As TimeSpan
Public TimeLeft As String

Private Sub ID_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    Counter1Red = Quota
    Counter2Blue = Quota
    Counter3Yellow = Quota

    'Determines maze color scheme to use for each cue
    IntMillisecond = Date.Now.Millisecond
    If IntMillisecond Mod 3 = 0 Then
        Color1 = 1 'red for cue 1
        Color2 = 2 'blue for cue 2
        Color3 = 3 'yellow for cue 3
    ElseIf IntMillisecond Mod 3 = 1 Then
        Color1 = 2 'blue for cue 1
        Color2 = 3 'yellow for cue 2
        Color3 = 1 'red for cue 3
    Else
        Color1 = 3 'yellow for cue 1
        Color2 = 1 'red for cue 2
        Color3 = 2 'blue for cue 3
    End If

    'Comment out these to have the ID screen appear
    'txtID.Text = "110"
    'txtFeedback.Text = "0"
    'txtCompensation.Text = "1"
    'Button1_Click(sender, e)

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    ID = Val(txtID.Text)
    Feedback = Val(txtFeedback.Text)
    Compensation = Val(txtCompensation.Text)
    If IsNumeric(txtID.Text) Then
        If ID <= 0 Then
            MsgBox("Please enter a valid participant ID.")
            txtID.Text = ""
            txtID.Select()
        ElseIf IsNumeric(txtFeedback.Text) Then
            If Feedback < 0 Then
                MsgBox("Please enter a valid feedback condition.")
                txtFeedback.Text = ""
                txtFeedback.Select()
            ElseIf Feedback > 3 Then
                MsgBox("Please enter a valid feedback condition.")
```

```
            txtFeedback.Text = ""
            txtFeedback.Select()
        ElseIf IsNumeric(txtCompensation.Text) Then
            If Compensation < 1 Then
                MsgBox("Please enter a valid compensation condition.")
                txtCompensation.Text = ""
                txtCompensation.Select()
            ElseIf Compensation > 2 Then
                MsgBox("Please enter a valid compensation condition.")
                txtCompensation.Text = ""
                txtCompensation.Select()
            Else
                'AlarmTime = DateAdd(DateInterval.Minute, Min1, Date.Now)
                'Yellow_Room1.Show()
                'Timer1.Enabled = True
                dlgGeneral.Show()
                Me.Hide()
            End If
        Else
            MsgBox("Please enter a numeric compensation condition.")
            txtCompensation.Text = ""
            txtCompensation.Select()
        End If
    Else
        MsgBox("Please enter a numeric feedback condition.")
        txtFeedback.Text = ""
        txtFeedback.Select()
    End If
  Else
    MsgBox("Please enter a numeric participant ID.")
    txtID.Text = ""
    txtID.Select()
  End If
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
  If AlarmTime < Date.Now Then
    EndingProcedure()
  Else
    RemainingTime = AlarmTime.Subtract(Date.Now)
    TimeLeft = RemainingTime.Minutes & ":" & Format(RemainingTime.Seconds, "00")
    Yellow_Room1.Label4.Text = TimeLeft
    Yellow_Room2.Label4.Text = TimeLeft
    Yellow_Room3.Label4.Text = TimeLeft
    Treasure_Room.Label4.Text = TimeLeft
    Orange_Room11.Label4.Text = TimeLeft
    Orange_Room12.Label4.Text = TimeLeft
    Orange_Room21.Label4.Text = TimeLeft
    Orange_Room22.Label4.Text = TimeLeft
    Orange_Room31.Label4.Text = TimeLeft
    Orange_Room32.Label4.Text = TimeLeft
    Red_Room111.Label4.Text = TimeLeft
    Red_Room112.Label4.Text = TimeLeft
    Red_Room113.Label4.Text = TimeLeft
    Red_Room121.Label4.Text = TimeLeft
    Red_Room122.Label4.Text = TimeLeft
    Red_Room123.Label4.Text = TimeLeft
    Red_Room211.Label4.Text = TimeLeft
    Red_Room212.Label4.Text = TimeLeft
    Red_Room213.Label4.Text = TimeLeft
    Red_Room221.Label4.Text = TimeLeft
    Red_Room222.Label4.Text = TimeLeft
    Red_Room223.Label4.Text = TimeLeft
```

```
            Red_Room311.Label4.Text = TimeLeft
            Red_Room312.Label4.Text = TimeLeft
            Red_Room313.Label4.Text = TimeLeft
            Red_Room321.Label4.Text = TimeLeft
            Red_Room322.Label4.Text = TimeLeft
            Red_Room323.Label4.Text = TimeLeft
            DarkRed_Room.Label4.Text = TimeLeft
        End If
    End Sub

    Public Sub EndingProcedure()
        Timer1.Stop()
        If Period = 1 Then
            Maze = Maze + 1
            SecComp1 = TotalSeconds1 * SecondRate
            If Compensation = 2 Then
                CashComp1 = FormatCurrency((Fixed + SecComp1), 2)
                Comp1 = Fixed + SecComp1
            ElseIf Compensation = 1 Then
                MazeComp1 = CompletedMaze1 * Rate
                CashComp1 = FormatCurrency((MazeComp1 + SecComp1), 2)
                Comp1 = MazeComp1 + SecComp1
            End If
        ElseIf Period = 2 Then
            SecComp2 = TotalSeconds2 * SecondRate
            If Compensation = 2 Then
                CashComp2 = FormatCurrency(((2 * Fixed) + SecComp2), 2)
                Comp2 = (2 * Fixed) + SecComp2
            ElseIf Compensation = 1 Then
                MazeComp2 = CompletedMaze2 * Rate
                CashComp2 = FormatCurrency((MazeComp2 + SecComp2), 2)
                Comp2 = MazeComp2 + SecComp2
            End If
        End If
        If Period = 1 Then
            UnloadAllForms()
            If CompletedMaze1 = 1 Then
                MsgBox("You completed " & CompletedMaze1 & " maze.  Your compensation for Round 1 is " & CashComp1 & ".",
MsgBoxStyle.OkOnly)
            ElseIf FinishedEarly1 = True Then
                MsgBox("You completed " & CompletedMaze1 & " mazes with " & TotalSeconds1 & " seconds remaining.  Your
compensation for Round 1 is " & CashComp1 & ".", MsgBoxStyle.OkOnly)
            Else
                MsgBox("You completed " & CompletedMaze1 & " mazes.  Your compensation for Round 1 is " & CashComp1 & ".",
MsgBoxStyle.OkOnly)
            End If
        ElseIf Period = 2 Then
            UnloadAllForms()
            If CompletedMaze2 = 1 Then
                MsgBox("You completed " & CompletedMaze2 & " maze.  Your compensation for Round 2 is " & CashComp2 & ".",
MsgBoxStyle.OkOnly)
            ElseIf FinishedEarly2 = True Then
                MsgBox("You completed " & CompletedMaze2 & " mazes with " & TotalSeconds2 & " seconds remaining.  Your
compensation for Round 2 is " & CashComp2 & ".", MsgBoxStyle.OkOnly)
            Else
                MsgBox("You completed " & CompletedMaze2 & " mazes.  Your compensation for Round 2 is " & CashComp2 & ".",
MsgBoxStyle.OkOnly)
            End If
        End If
        If Period = 1 Then
            QuestAttract1.Show()
            Me.Hide()
        ElseIf Period = 2 Then
```

```
        QuestAttract2.Show()
        Me.Hide()
      End If
    End Sub

    Private Sub txtID_TabStop(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtID.GotFocus
      txtID.SelectAll()
    End Sub
    Private Sub txtFeedback_TabStop(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtFeedback.GotFocus
      txtFeedback.SelectAll()
    End Sub
    Private Sub txtCompensation_TabStop(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
txtCompensation.GotFocus
      txtCompensation.SelectAll()
    End Sub
End Class
```

B.2 dlgGeneral.vb

```
Public Class dlgGeneral
    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
      dlgTask.Show()
      Me.Close()
    End Sub
End Class
```

B.3 dlgTask.vb

```
Public Class dlgTask
    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
      'If CbxAcknowledge.Checked = False Then
      'ShowMessage()
      'Else
      dlgTask_Cont1.Show()
      Me.Close()
      'End If
    End Sub
    Private Sub Back_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Back_Button.Click
      dlgGeneral.Show()
      Me.Close()
    End Sub
    Private Sub ShowMessage()
      MessageBox.Show("Please check the box indicating that you have read and understand this information.", "",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
End Class
```

B.4 dlgTask_Cont1.vb

```
Public Class dlgTask_Cont1
    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
      'If CbxAcknowledge.Checked = False Then
      'ShowMessage()
      'Else
      dlgTask_Cont2.Show()
      Me.Close()
      'End If
    End Sub
    Private Sub Back_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Back_Button.Click
```

```
        dlgTask.Show()
        Me.Close()
    End Sub
    Private Sub ShowMessage()
        MessageBox.Show("Please check the box indicating that you have read and understand this information.", "",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
End Class
```

B.4 dlgTask_Cont2.vb

```
Public Class dlgTask_Cont2

    Dim BooFirst As Boolean = False
    Dim BooSecond As Boolean = False
    Dim BooThird As Boolean = False

    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
        If BooFirst = True Then
            If BooSecond = True Then
                If BooThird = True Then
                    dlgTask_Cont3.Show()
                    Me.Close()
                Else
                    MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
                End If
            Else
                MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
            End If
        Else
            MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
        End If
    End Sub

    Private Sub Back_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Back_Button.Click
        dlgTask_Cont1.Show()
        Me.Close()
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.Click
        BooFirst = True
    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.Click
        showmessage1()
        BooFirst = False
    End Sub

    Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton4.Click
        showmessage2()
        BooSecond = False
    End Sub

    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.Click
        BooSecond = True
    End Sub
```

```
    Private Sub RadioButton5_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton5.Click
        BooThird = True
    End Sub

    Private Sub RadioButton6_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton6.Click
        showmessage3()
        BooThird = False
    End Sub

    Private Sub showmessage1()
        MessageBox.Show("Incorrect. The answer is TRUE. The same cues will be used in the same maze types (e.g. red, yellow, or
blue) for both the first and second rounds.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub

    Private Sub showmessage2()
        MessageBox.Show("Incorrect. The answer is FALSE. Cues are NOT related to the positioning of objects in each room, but only
consist of the presence or absence of a certain object or objects in the rooms.", "Incorrect", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    End Sub

    Private Sub showmessage3()
        MessageBox.Show("Incorrect. The answer is FALSE. Cues indicate that you are on INCORRECT paths.", "Incorrect",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
End Class
```

B.5 dlgTask_Cont3.vb

```
Public Class dlgTask_Cont3
    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
        'If CbxAcknowledge.Checked = False Then
        'ShowMessage()
        'Else
        QuestAttract.Show()
        Me.Close()
        'End If
    End Sub
    Private Sub Back_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Back_Button.Click
        dlgTask_Cont2.Show()
        Me.Close()
    End Sub
    Private Sub ShowMessage()
        MessageBox.Show("Please check the box indicating that you have read and understand this information.", "",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
    Private Sub dlgTask_Cont3_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        RichTextBox1.Text = "TASK (Continued)" & vbCrLf & vbCrLf & "In the FIRST round, you will be given the same type of
maze (e.g. red, yellow, or blue) for three consecutive mazes followed by the second type of maze for the next three mazes followed by
the third type of maze for the next three mazes.  After these nine mazes, you will be given a different maze type (alternating between
the three - red, yellow, and blue) for each maze until the time runs out." & vbCrLf & vbCrLf & _
                    "In the SECOND round, you will be given the choice of which type of maze (e.g. red, yellow, or blue) to complete –
up to a predetermined number of mazes for each type.  At the end of each maze, you will be given the same choice unless you have
previously completed all of the allotted mazes of a particular type (i.e. color), in which case you will be allowed to choose from mazes
of the remaining type(s).  For example, if you complete all of the allotted yellow-doored mazes, you would then only be allowed to
choose a red- or a blue-doored maze.  After completing all of the allotted mazes for each type (i.e. color), each subsequent maze will
be of a randomly determined maze type (between the three available)." & vbCrLf & vbCrLf & _
                    "You will work on the maze task for " & ID.Min1 & " minutes in the first round and " & ID.Min2 & " minutes in the
second round.  There is a maximum of " & ID.Max1 & " mazes to complete in the first round and " & ID.Max2 & " mazes to complete
in the second round."
    End Sub
```

```
End Class
```

B.6 QuestAttract.vb (essentially the same code for QuestAttract1.vb and QuestAttract2.vb)

```
Public Class QuestAttract
    Public nOne As Integer
    Public nTwo As Integer
    Public nThree As Integer
    Public nFour As Integer
    Public nFive As Integer
    Public nSix As Integer
    Public nSeven As Integer
    Private Sub btnSubmit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click
        If rb1_7.Checked = False Then
            If rb1_6.Checked = False Then
                If rb1_5.Checked = False Then
                    If rb1_4.Checked = False Then
                        If rb1_3.Checked = False Then
                            If rb1_2.Checked = False Then
                                If rb1_1.Checked = False Then
                                    MsgBox("Please select a response for 'Attractive/Repulsive' scale.  Click OK to continue.")
                                    nOne = 0
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
        If rb1_7.Checked = True Then
            nOne = 7
        ElseIf rb1_6.Checked = True Then
            nOne = 6
        ElseIf rb1_5.Checked = True Then
            nOne = 5
        ElseIf rb1_4.Checked = True Then
            nOne = 4
        ElseIf rb1_3.Checked = True Then
            nOne = 3
        ElseIf rb1_2.Checked = True Then
            nOne = 2
        ElseIf rb1_1.Checked = True Then
            nOne = 1
        End If
        If rb2_7.Checked = False Then
            If rb2_6.Checked = False Then
                If rb2_5.Checked = False Then
                    If rb2_4.Checked = False Then
                        If rb2_3.Checked = False Then
                            If rb2_2.Checked = False Then
                                If rb2_1.Checked = False Then
                                    MsgBox("Please select a response for 'Dull/Exciting' scale.  Click OK to continue.")
                                    nTwo = 0
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
        If rb2_7.Checked = True Then
```

```
    nTwo = 7
ElseIf rb2_6.Checked = True Then
    nTwo = 6
ElseIf rb2_5.Checked = True Then
    nTwo = 5
ElseIf rb2_4.Checked = True Then
    nTwo = 4
ElseIf rb2_3.Checked = True Then
    nTwo = 3
ElseIf rb2_2.Checked = True Then
    nTwo = 2
ElseIf rb2_1.Checked = True Then
    nTwo = 1
End If
If rb3_7.Checked = False Then
    If rb3_6.Checked = False Then
        If rb3_5.Checked = False Then
            If rb3_4.Checked = False Then
                If rb3_3.Checked = False Then
                    If rb3_2.Checked = False Then
                        If rb3_1.Checked = False Then
                            MsgBox("Please select a response for 'Good/Bad' scale.  Click OK to continue.")
                            nThree = 0
                        End If
                    End If
                End If
            End If
        End If
    End If
End If
If rb3_7.Checked = True Then
    nThree = 7
ElseIf rb3_6.Checked = True Then
    nThree = 6
ElseIf rb3_5.Checked = True Then
    nThree = 5
ElseIf rb3_4.Checked = True Then
    nThree = 4
ElseIf rb3_3.Checked = True Then
    nThree = 3
ElseIf rb3_2.Checked = True Then
    nThree = 2
ElseIf rb3_1.Checked = True Then
    nThree = 1
End If
If rb4_7.Checked = False Then
    If rb4_6.Checked = False Then
        If rb4_5.Checked = False Then
            If rb4_4.Checked = False Then
                If rb4_3.Checked = False Then
                    If rb4_2.Checked = False Then
                        If rb4_1.Checked = False Then
                            MsgBox("Please select a response for 'Boring/Interesting' scale.  Click OK to continue.")
                            nFour = 0
                        End If
                    End If
                End If
            End If
        End If
    End If
End If
If rb4_7.Checked = True Then
    nFour = 7
```

```
ElseIf rb4_6.Checked = True Then
   nFour = 6
ElseIf rb4_5.Checked = True Then
   nFour = 5
ElseIf rb4_4.Checked = True Then
   nFour = 4
ElseIf rb4_3.Checked = True Then
   nFour = 3
ElseIf rb4_2.Checked = True Then
   nFour = 2
ElseIf rb4_1.Checked = True Then
   nFour = 1
End If
If rb5_7.Checked = False Then
   If rb5_6.Checked = False Then
      If rb5_5.Checked = False Then
         If rb5_4.Checked = False Then
            If rb5_3.Checked = False Then
               If rb5_2.Checked = False Then
                  If rb5_1.Checked = False Then
                     MsgBox("Please select a response for 'Superior/Inferior' scale.  Click OK to continue.")
                     nFive = 0
                  End If
               End If
            End If
         End If
      End If
   End If
End If
If rb5_7.Checked = True Then
   nFive = 7
ElseIf rb5_6.Checked = True Then
   nFive = 6
ElseIf rb5_5.Checked = True Then
   nFive = 5
ElseIf rb5_4.Checked = True Then
   nFive = 4
ElseIf rb5_3.Checked = True Then
   nFive = 3
ElseIf rb5_2.Checked = True Then
   nFive = 2
ElseIf rb5_1.Checked = True Then
   nFive = 1
End If
If rb6_7.Checked = False Then
   If rb6_6.Checked = False Then
      If rb6_5.Checked = False Then
         If rb6_4.Checked = False Then
            If rb6_3.Checked = False Then
               If rb6_2.Checked = False Then
                  If rb6_1.Checked = False Then
                     MsgBox("Please select a response for 'Unwholesome/Wholesome' scale.  Click OK to continue.")
                     nSix = 0
                  End If
               End If
            End If
         End If
      End If
   End If
End If
If rb6_7.Checked = True Then
   nSix = 7
ElseIf rb6_6.Checked = True Then
```

```
        nSix = 6
    ElseIf rb6_5.Checked = True Then
        nSix = 5
    ElseIf rb6_4.Checked = True Then
        nSix = 4
    ElseIf rb6_3.Checked = True Then
        nSix = 3
    ElseIf rb6_2.Checked = True Then
        nSix = 2
    ElseIf rb6_1.Checked = True Then
        nSix = 1
    End If
    If rb7_7.Checked = False Then
        If rb7_6.Checked = False Then
            If rb7_5.Checked = False Then
                If rb7_4.Checked = False Then
                    If rb7_3.Checked = False Then
                        If rb7_2.Checked = False Then
                            If rb7_1.Checked = False Then
                                MsgBox("Please select a response for 'Fun/Tedious' scale.  Click OK to continue.")
                                nSeven = 0
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
    If rb7_7.Checked = True Then
        nSeven = 7
    ElseIf rb7_6.Checked = True Then
        nSeven = 6
    ElseIf rb7_5.Checked = True Then
        nSeven = 5
    ElseIf rb7_4.Checked = True Then
        nSeven = 4
    ElseIf rb7_3.Checked = True Then
        nSeven = 3
    ElseIf rb7_2.Checked = True Then
        nSeven = 2
    ElseIf rb7_1.Checked = True Then
        nSeven = 1
    End If
    If nOne > 0 Then
        If nTwo > 0 Then
            If nThree > 0 Then
                If nFour > 0 Then
                    If nFive > 0 Then
                        If nSix > 0 Then
                            If nSeven > 0 Then
                                dlgFeedback.Show()
                                Me.Hide()
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
End Sub
Private Sub frmQuestAttract_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    MsgBox("Please complete the following questionnaire. Click OK to continue.", , "")
    rb1_7.Checked = False
```

```
        rb1_7.TabStop = True
    End Sub
End Class
```

B.7 dlgFeedback.vb

```
Public Class dlgFeedback

    Dim BooFirst As Boolean = False
    Dim BooSecond As Boolean = False

    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
        If BooFirst = True Then
            If BooSecond = True Then
                dlgCompensation.Show()
                Me.Close()
            Else
                MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
            End If
        Else
            MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
        End If
    End Sub

    Private Sub dlgFeedback_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        If ID.Feedback = 0 Then
            RichTextBox1.Text = "During the first round, you will immediately be notified whether or not a selected path is incorrect.  An example of the notification message is seen below.  However, you may proceed down incorrect paths for as long as you would like.  During the second round, you will NOT be notified that a selected path is incorrect."
            Label1.Text = "During the first round, you will be notified immediately whether or not a selected path is incorrect."
            Label2.Text = "During the second round, you will be notified whether or not a selected path is incorrect."
        ElseIf ID.Feedback = 1 Then
            RichTextBox1.Text = "During the first round, you will immediately be notified once you have entered an incorrect path.  An example of the notification message is seen below.  However, you may continue down incorrect paths for as long as you would like.  During the second round, you will NOT be notified that you have entered an incorrect path."
            Label1.Text = "During the first round, you will be notified immediately that you have entered an incorrect path."
        ElseIf ID.Feedback = 2 Then
            RichTextBox1.Text = "During the first round, you will be notified that you have entered an incorrect path after a one-room delay.  An example of the notification message is seen below.  However, you may continue down incorrect paths for as long as you would like.  During the second round, you will NOT be notified that you have entered an incorrect path."
            Label1.Text = "During the first round, you will be notified after a one-room delay that you have entered an incorrect path."
        ElseIf ID.Feedback = 3 Then
            RichTextBox1.Text = "During the first round, you will be notified that you have entered an incorrect path after a two-room delay (i.e. at dead ends).  An example of the notification message is seen below.  However, you may continue in incorrect paths for as long as you would like.  During the second round, you will NOT be notified that you have entered an incorrect path."
            Label1.Text = "During the first round, you will be notified after a two-room delay (i.e. at dead ends) that you have entered an incorrect path."
        End If
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.Click
        BooFirst = True
    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.Click
        showmessage1()
        BooFirst = False
    End Sub
```

```
    Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton4.Click
        showmessage2()
        BooSecond = False
    End Sub

    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.Click
        BooSecond = True
    End Sub

    Private Sub showmessage1()
        If ID.Feedback = 0 Then
            MessageBox.Show("Incorrect. The answer is TRUE. During the first round, you will be notified immediately whether or not a
selected path is incorrect.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
        ElseIf ID.Feedback = 1 Then
            MessageBox.Show("Incorrect. The answer is TRUE. During the first round, you will be notified immediately that you have
entered an incorrect path.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
        ElseIf ID.Feedback = 2 Then
            MessageBox.Show("Incorrect. The answer is TRUE. During the first round, you will be notified after a one-room delay that
you have entered an incorrect path.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
        ElseIf ID.Feedback = 3 Then
            MessageBox.Show("Incorrect. The answer is TRUE. During the first round, you will be notified after a two-room delay (i.e. at
dead ends) that you have entered an incorrect path.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub

    Private Sub showmessage2()
        If ID.Feedback = 0 Then
            MessageBox.Show("Incorrect. The answer is FALSE. During the second round, you will NOT be notified whether or not the
selected path is incorrect.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
            MessageBox.Show("Incorrect. The answer is FALSE. During the second round, you will NOT be notified that you have
entered an incorrect path.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub

End Class
```

B.8 dlgCompensation.vb

```
Public Class dlgCompensation
    Dim Third As Single
    Dim Fourth As Single
    Dim Three As Single
    Dim Four As Single
    Dim Comp3 As String
    Dim Comp4 As String
    Dim Rate As String
    Dim SecondRate As String
    Dim SecExample As String
    Dim Fixed1 As String
    Dim Fixed2 As String
    Dim BooFirst As Boolean = False
    Dim BooSecond As Boolean = False
    Dim BooThird As Boolean = False
    Dim BooFourth As Boolean = False

    Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK_Button.Click
        If BooFirst = True Then
            If BooSecond = True Then
                If BooThird = True Then
```

```
            If BooFourth = True Then
               MsgBox("Click "OK" when you are ready to begin.", , "")
               ID.AlarmTime = DateAdd(DateInterval.Minute, ID.Min1, Date.Now)
               Yellow_Room1.Show()
               ID.Timer1.Enabled = True
               Me.Close()
            Else
               MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
            End If
         Else
            MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
         End If
      Else
         MsgBox("Please ensure that you have the selected the correct reponses. Click OK to continue.")
      End If
   End Sub

   Private Sub dlgCompensation_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
      Fixed1 = FormatCurrency((ID.Fixed), 2)
      Fixed2 = FormatCurrency((ID.Fixed * 2), 2)
      Rate = FormatCurrency((ID.Rate), 2)
      SecondRate = FormatCurrency((ID.SecondRate), 2)
      SecExample = FormatCurrency((ID.SecondRate * 60), 2)
      If ID.Compensation = 1 Then
         RichTextBox1.Text = "The number of mazes you complete is highly valued, particularly during the second round.  Every participant's anonymous performance in the second round will be posted following the session." & vbCrLf & vbCrLf & "Your compensation will be based on HOW MANY mazes you complete (i.e. by reaching each 'Treasure Room') in both the first and second rounds.  You will be paid " & Rate & " in both the first and second rounds for each maze that you complete in the given time (up to a maximum of " & ID.Max1 & " in the first round and " & ID.Max2 & " in the second round).  The more mazes you complete, the more money you will make.  Note that the time given to complete mazes in the second round is twice the time given to complete mazes in the first round.  All participants in this session will receive " & Rate & " for each maze completed in the allotted time for each round in cash at the end of the experiment." & vbCrLf & vbCrLf & "In the event that you complete all of the allotted mazes for a given round, you will be paid a bonus of " & SecondRate & " per second remaining for that round.  For example, if you complete all of the mazes in the first round with 1 minute remaining you will receive a bonus of " & SecExample & " (60 seconds * " & ID.SecondRate & ")."
      ElseIf ID.Compensation = 2 Then
         RichTextBox1.Text = "The number of mazes you complete is highly valued, particularly during the second round.  Every participant's anonymous performance in the second round will be posted following the session." & vbCrLf & vbCrLf & "You will receive a fixed payment for working on the maze task in each round, regardless of the number of mazes you complete (up to a maximum of " & ID.Max1 & " in the first round and " & ID.Max2 & " in the second round).  All participants in this session will receive a fixed payment of " & Fixed1 & " for participating in the first round and " & Fixed2 & " for participating in the second round in cash at the conclusion of this experiment.  Note that the time given to complete mazes in the second round is twice the time given to complete mazes in the first round." & vbCrLf & vbCrLf & "In the event that you complete all of the allotted mazes for a given round, you will be paid a bonus of " & SecondRate & " per second remaining for that round.  For example, if you complete all of the mazes in the first round with 1 minute remaining you will receive a bonus of " & SecExample & " (60 seconds * " & ID.SecondRate & ")."
      End If
   End Sub

   Private Sub txtThird_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtThird.Leave
      Third = 10 * Val(txtThird.Text)
      Three = 10 * 3 * ID.Rate
      If ID.Compensation = 1 Then
         Comp3 = FormatCurrency((3 * ID.Rate), 2)
         If Third <> Three Then
            showmessage3()
            txtThird.Text = ""
            txtThird.Focus()
         Else
            BooThird = True
         End If
```

```
        ElseIf ID.Compensation = 2 Then
            Comp4 = FormatCurrency((ID.Fixed * 3), 2)
            If Third <> (ID.Fixed * 3 * 10) Then
                showmessage4()
                txtThird.Text = ""
                txtThird.Focus()
            Else
                BooThird = True
            End If
        End If
    End Sub

    Private Sub txtFourth_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtFourth.Leave
        Fourth = 10 * Val(txtFourth.Text)
        Four = 10 * 60 * ID.Rate
        If ID.Compensation = 1 Then
            Comp4 = FormatCurrency((60 * ID.Rate), 2)
            If Fourth <> Four Then
                showmessage5()
                txtFourth.Text = ""
                txtFourth.Focus()
            Else
                BooFourth = True
            End If
        ElseIf ID.Compensation = 2 Then
            Comp4 = FormatCurrency((ID.Fixed * 3), 2)
            If Fourth <> (ID.Fixed * 3 * 10) Then
                showmessage4()
                txtFourth.Text = ""
                txtFourth.Focus()
            Else
                BooFourth = True
            End If
        End If
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.Click
        BooFirst = True
    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.Click
        showmessage1()
        BooFirst = False
    End Sub

    Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton4.Click
        showmessage2()
        BooSecond = False
    End Sub

    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.Click
        BooSecond = True
    End Sub

    Private Sub showmessage1()
        MessageBox.Show("Incorrect. The answer is TRUE. There is a maximum number of mazes you can complete in each round (20
in the first round and 40 in the second round).", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
```

```
    Private Sub showmessage2()
        MessageBox.Show("Incorrect. The answer is FALSE. You will receive a bonus of " & SecondRate & " for each second
remaining when you complete all of the allotted mazes for a given round.", "Incorrect", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    End Sub

    Private Sub showmessage3()
        MessageBox.Show("Incorrect. Your total compensation would be " & Comp3 & " [" & Rate & " * 3], where " & Rate & " is the
piece-rate for each completed maze for each round.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub

    Private Sub showmessage4()
        MessageBox.Show("Incorrect. Your compensation would be " & Comp4 & " [" & Fixed1 & " + " & Fixed2 & "], where " &
Fixed1 & " is the fixed wage for first round and " & Fixed2 & " is the fixed wage for the second round.", "Incorrect",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub

    Private Sub showmessage5()
        MessageBox.Show("Incorrect. Your total compensation would be " & Comp4 & " [" & Rate & " * 60], where " & Rate & " is the
piece-rate for each completed maze for each round.", "Incorrect", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub

    Private Sub Back_Button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Back_Button.Click
        dlgFeedback.Show()
        Me.Close()
    End Sub
End Class
```

B.9 Yellow_Room1.vb (Y_R2.vb and Y_R3.vb essentially the same)

```
Public Class Yellow_Room1

    Dim Period As Integer
    Dim Room As Single
    Dim Maze As Integer
    Public Direction As Integer
    Dim ImageGroup As Integer
    Public Correct1 As Single
    Dim IntMillisecond As Integer
    Dim Int1 As Integer
    Dim Int2 As Integer
    Dim Int3 As Integer
    Dim Int4 As Integer
    Dim Int5 As Integer
    Dim Int6 As Integer
    Dim Int7 As Integer
    Dim Int8 As Integer
    Dim Int9 As Integer
    Dim Progress As Integer
    Public FeedbackGiven1 As Boolean
    Public FeedbackGiven2 As Boolean
    Public FeedbackGiven3 As Boolean
    Public AddNumber As Integer = 5
    Public AddNumber1 As Single

    Private Sub Yellow_Room1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv",
"Participant,Feedback,Compensation,Period,Maze,Cue,Room,Time,Millisecond,Direction,Correct1,Correct2,Correct3" & vbCrLf,
True)
        IntMillisecond = Date.Now.Millisecond
        Period = ID.Period
        Maze = ID.Maze
```

```
Room = 1
Correct1 = IntMillisecond Mod 3

'Determine images for PictureBoxes

Int1 = CInt(Rnd() * IntMillisecond)
Int2 = CInt(Rnd() * IntMillisecond)
Int3 = CInt(Rnd() * IntMillisecond)
Int4 = CInt(Rnd() * IntMillisecond)
Int5 = CInt(Rnd() * IntMillisecond)
Int6 = CInt(Rnd() * IntMillisecond)
Int7 = CInt(Rnd() * IntMillisecond)
Int8 = CInt(Rnd() * IntMillisecond)
Int9 = CInt(Rnd() * IntMillisecond)

'Sets time delay for the immediate, short-delayed, and long-delayed conditions the same to get into the incorrect room
If Period = 1 Then
   If ID.Feedback > 0 Then
      AddNumber1 = 5
   Else
      AddNumber1 = 10
   End If
Else
   AddNumber1 = 5
End If

If ID.Cue = 1 Then
   "Color of maze is put in place
   If ID.Color1 = 1 Then
      PictureBox13.Image = My.Resources.Room1
   ElseIf ID.Color1 = 2 Then
      PictureBox13.Image = My.Resources.Room2
   Else
      PictureBox13.Image = My.Resources.Room3
   End If
   "Ensures non-overlapping of images for corners and wall images
   If Int6 Mod 3 <> 0 Then
      Int7 = 3
   End If
   If Int8 Mod 3 <> 0 Then
      Int5 = 3
   End If
   'Presence/absence of portrait is the cue
   If Int1 Mod 3 = 1 Then
      PictureBox1.Image = My.Resources.Torch1
      PictureBox1.Visible = True
   ElseIf Int1 Mod 3 = 2 Then
      PictureBox1.Image = My.Resources.Torch3
      PictureBox1.Visible = True
   End If
   If Int2 Mod 3 = 1 Then
      PictureBox2.Image = My.Resources.Torch
      PictureBox2.Visible = True
   ElseIf Int2 Mod 3 = 2 Then
      PictureBox2.Image = My.Resources.Torch2
      PictureBox2.Visible = True
   End If
   If Int3 Mod 4 = 1 Then
      PictureBox3.Image = My.Resources.deer_head
      PictureBox3.Visible = True
   ElseIf Int3 Mod 4 = 2 Then
      PictureBox3.Image = My.Resources.Picture2
      PictureBox3.Width = 129
```

```
    PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 4 = 3 Then
    PictureBox3.Image = My.Resources.Picture3
    PictureBox3.Width = 108
    PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
    PictureBox3.Visible = True
End If
If Int4 Mod 4 = 1 Then
    PictureBox4.Image = My.Resources.deer_head
    PictureBox4.Visible = True
ElseIf Int4 Mod 4 = 2 Then
    PictureBox4.Image = My.Resources.Picture2
    PictureBox4.Width = 129
    PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
    PictureBox4.Visible = True
ElseIf Int4 Mod 4 = 3 Then
    PictureBox4.Image = My.Resources.Picture3
    PictureBox4.Width = 108
    PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
    PictureBox4.Visible = True
End If
If Int5 Mod 3 = 1 Then
    PictureBox5.Image = My.Resources.Clock2
    PictureBox5.Height = 200
    PictureBox5.Width = 59
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
ElseIf Int5 Mod 3 = 2 Then
    PictureBox5.Image = My.Resources.masters_chair
    PictureBox5.Height = 200
    PictureBox5.Width = 132
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
End If
If Int6 Mod 3 = 1 Then
    PictureBox6.Image = My.Resources.Clock1
    PictureBox6.Height = 200
    PictureBox6.Width = 59
    PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
    PictureBox6.Visible = True
ElseIf Int6 Mod 3 = 2 Then
    PictureBox6.Image = My.Resources.masters_chair1
    PictureBox6.Height = 200
    PictureBox6.Width = 132
    PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
    PictureBox6.Visible = True
End If
If Int7 Mod 3 = 1 Then
    PictureBox7.Image = My.Resources.Secretaire
    PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
    PictureBox7.Visible = True
ElseIf Int7 Mod 3 = 2 Then
    PictureBox7.Image = My.Resources.Fireplace
    PictureBox7.Height = 133
    PictureBox7.Width = 153
    PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
    PictureBox7.Visible = True
End If
If Int8 Mod 3 = 1 Then
    PictureBox8.Image = My.Resources.Secretaire1
    PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
    PictureBox8.Visible = True
```

```
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    ElseIf ID.Cue = 2 Then
        "Color of maze is put in place
        If ID.Color2 = 1 Then
            PictureBox13.Image = My.Resources.Room1
        ElseIf ID.Color2 = 2 Then
            PictureBox13.Image = My.Resources.Room2
        Else
            PictureBox13.Image = My.Resources.Room3
        End If
        "Ensures non-overlapping of images for corners and wall images
        If Int6 Mod 3 <> 0 Then
            Int7 = 3
        End If
        If Int8 Mod 3 <> 0 Then
            Int5 = 3
        End If
        'Type of torch is the cue
        If Int1 Mod 2 = 1 Then
            PictureBox1.Image = My.Resources.Torch1
            PictureBox1.Visible = True
        End If
        If Int2 Mod 2 = 1 Then
            PictureBox2.Image = My.Resources.Torch
            PictureBox2.Visible = True
        End If
        If Int3 Mod 5 = 1 Then
            PictureBox3.Image = My.Resources.deer_head
            PictureBox3.Visible = True
        ElseIf Int3 Mod 5 = 2 Then
            PictureBox3.Image = My.Resources.Picture2
            PictureBox3.Width = 129
            PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
            PictureBox3.Visible = True
        ElseIf Int3 Mod 5 = 3 Then
            PictureBox3.Image = My.Resources.Picture3
            PictureBox3.Width = 108
            PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
            PictureBox3.Visible = True
        ElseIf Int3 Mod 5 = 4 Then
            PictureBox3.Image = My.Resources.Portrait
            PictureBox3.Visible = True
        End If
        If Int4 Mod 5 = 1 Then
            PictureBox4.Image = My.Resources.deer_head
            PictureBox4.Visible = True
        ElseIf Int4 Mod 5 = 3 Then
            PictureBox4.Image = My.Resources.Picture2
            PictureBox4.Width = 129
            PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
```

```
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 3 Then
    PictureBox4.Image = My.Resources.Picture3
    PictureBox4.Width = 108
    PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 4 Then
    PictureBox4.Image = My.Resources.Portrait
    PictureBox4.Visible = True
End If
If Int5 Mod 3 = 1 Then
    PictureBox5.Image = My.Resources.Clock2
    PictureBox5.Height = 200
    PictureBox5.Width = 59
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
ElseIf Int5 Mod 3 = 2 Then
    PictureBox5.Image = My.Resources.masters_chair
    PictureBox5.Height = 200
    PictureBox5.Width = 132
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
End If
If Int6 Mod 3 = 1 Then
    PictureBox6.Image = My.Resources.Clock1
    PictureBox6.Height = 200
    PictureBox6.Width = 59
    PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
    PictureBox6.Visible = True
ElseIf Int6 Mod 3 = 2 Then
    PictureBox6.Image = My.Resources.masters_chair1
    PictureBox6.Height = 200
    PictureBox6.Width = 132
    PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
    PictureBox6.Visible = True
End If
If Int7 Mod 3 = 1 Then
    PictureBox7.Image = My.Resources.Secretaire
    PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
    PictureBox7.Visible = True
ElseIf Int7 Mod 3 = 2 Then
    PictureBox7.Image = My.Resources.Fireplace
    PictureBox7.Height = 133
    PictureBox7.Width = 153
    PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
    PictureBox7.Visible = True
End If
If Int8 Mod 3 = 1 Then
    PictureBox8.Image = My.Resources.Secretaire1
    PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
    PictureBox8.Visible = True
ElseIf Int8 Mod 3 = 2 Then
    PictureBox8.Image = My.Resources.Fireplace
    PictureBox8.Height = 133
    PictureBox8.Width = 153
    PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
    PictureBox8.Visible = True
End If
If Int9 Mod 3 = 1 Then
    PictureBox9.Image = My.Resources.rug1
    PictureBox9.Visible = True
ElseIf Int9 Mod 3 = 2 Then
    PictureBox9.Image = My.Resources.rug2
```

```
            PictureBox9.Visible = True
        End If
    Else
        "Color of maze is put in place
        If ID.Color3 = 1 Then
            PictureBox13.Image = My.Resources.Room1
        ElseIf ID.Color3 = 2 Then
            PictureBox13.Image = My.Resources.Room2
        Else
            PictureBox13.Image = My.Resources.Room3
        End If
        "Ensures non-overlapping of images for corners and wall images
        If Int6 Mod 2 <> 0 Then
            Int7 = 3
        End If
        If Int8 Mod 3 <> 0 Then
            Int5 = 2
        End If
        'Presence/absence of chair(s) is cue
        If Int1 Mod 3 = 1 Then
            PictureBox1.Image = My.Resources.Torch1
            PictureBox1.Visible = True
        ElseIf Int1 Mod 3 = 2 Then
            PictureBox1.Image = My.Resources.Torch3
            PictureBox1.Visible = True
        End If
        If Int2 Mod 3 = 1 Then
            PictureBox2.Image = My.Resources.Torch
            PictureBox2.Visible = True
        ElseIf Int2 Mod 3 = 2 Then
            PictureBox2.Image = My.Resources.Torch2
            PictureBox2.Visible = True
        End If
        If Int3 Mod 5 = 1 Then
            PictureBox3.Image = My.Resources.deer_head
            PictureBox3.Visible = True
        ElseIf Int3 Mod 5 = 2 Then
            PictureBox3.Image = My.Resources.Picture2
            PictureBox3.Width = 129
            PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
            PictureBox3.Visible = True
        ElseIf Int3 Mod 5 = 3 Then
            PictureBox3.Image = My.Resources.Picture3
            PictureBox3.Width = 108
            PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
            PictureBox3.Visible = True
        ElseIf Int3 Mod 5 = 4 Then
            PictureBox3.Image = My.Resources.Portrait
            PictureBox3.Visible = True
        End If
        If Int4 Mod 5 = 1 Then
            PictureBox4.Image = My.Resources.deer_head
            PictureBox4.Visible = True
        ElseIf Int4 Mod 5 = 2 Then
            PictureBox4.Image = My.Resources.Picture2
            PictureBox4.Width = 129
            PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
            PictureBox4.Visible = True
        ElseIf Int4 Mod 5 = 3 Then
            PictureBox4.Image = My.Resources.Picture3
            PictureBox4.Width = 108
            PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
            PictureBox4.Visible = True
```

```
        ElseIf Int4 Mod 5 = 4 Then
            PictureBox4.Image = My.Resources.Portrait
            PictureBox4.Visible = True
        End If
        If Int5 Mod 2 = 1 Then
            PictureBox5.Image = My.Resources.Clock2
            PictureBox5.Height = 200
            PictureBox5.Width = 59
            PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
            PictureBox5.Visible = True
        End If
        If Int6 Mod 2 = 1 Then
            PictureBox6.Image = My.Resources.Clock1
            PictureBox6.Height = 200
            PictureBox6.Width = 59
            PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        End If
        If Int7 Mod 3 = 1 Then
            PictureBox7.Image = My.Resources.Secretaire
            PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
            PictureBox7.Visible = True
        ElseIf Int7 Mod 3 = 2 Then
            PictureBox7.Image = My.Resources.Fireplace
            PictureBox7.Height = 133
            PictureBox7.Width = 153
            PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
            PictureBox7.Visible = True
        End If
        If Int8 Mod 3 = 1 Then
            PictureBox8.Image = My.Resources.Secretaire1
            PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
            PictureBox8.Visible = True
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    End If

    'MsgBox("The variable 'Correct' is " & Correct1 & "!", MsgBoxStyle.Exclamation, )
    CountForm(Me)
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & "," &
Correct1 & vbCrLf, True)
  End Sub

  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Direction = 1
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
    If Period = 1 Then
      If ID.Feedback = 0 Then
```

```
            If Correct1 = 0 Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            ElseIf FeedbackGiven1 = True Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            Else
                Timer2.Enabled = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                FeedbackGiven1 = True
            End If
        ElseIf ID.Feedback = 1 Then
            If Correct1 = 0 Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            Else
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
                FeedbackGiven1 = True
            End If
        Else
            ProgressBar1.Visible = True
            Label2.Visible = True
            Button1.Enabled = False
            Button2.Enabled = False
            Button3.Enabled = False
            Timer1.Enabled = True
        End If
    Else
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Timer1.Enabled = True
    End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Direction = 2
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
    If Period = 1 Then
        If ID.Feedback = 0 Then
```

```
            If Correct1 = 1 Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            ElseIf FeedbackGiven2 = True Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            Else
                Timer2.Enabled = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                FeedbackGiven2 = True
            End If
        ElseIf ID.Feedback = 1 Then
            If Correct1 = 1 Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            Else
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
                FeedbackGiven2 = True
            End If
        Else
            ProgressBar1.Visible = True
            Label2.Visible = True
            Button1.Enabled = False
            Button2.Enabled = False
            Button3.Enabled = False
            Timer1.Enabled = True
        End If
    Else
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Timer1.Enabled = True
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Direction = 3
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
    If Period = 1 Then
        If ID.Feedback = 0 Then
```

```
            If Correct1 = 2 Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            ElseIf FeedbackGiven3 = True Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            Else
                Timer2.Enabled = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                FeedbackGiven3 = True
            End If
        ElseIf ID.Feedback = 1 Then
            If Correct1 = 2 Then
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
            Else
                ProgressBar1.Visible = True
                Label2.Visible = True
                Button1.Enabled = False
                Button2.Enabled = False
                Button3.Enabled = False
                Timer1.Enabled = True
                FeedbackGiven3 = True
            End If
        Else
            ProgressBar1.Visible = True
            Label2.Visible = True
            Button1.Enabled = False
            Button2.Enabled = False
            Button3.Enabled = False
            Timer1.Enabled = True
        End If
    Else
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Timer1.Enabled = True
    End If
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Value = Progress
    If Progress < 100 Then
        Progress = Progress + AddNumber1
    ElseIf Progress = 100 Then
        If Direction = 1 Then
            If Correct1 = 0 Then
```

```
            Yellow_Room2.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        Else
            Orange_Room11.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        End If
    ElseIf Direction = 2 Then
        If Correct1 = 1 Then
            Yellow_Room2.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        ElseIf Correct1 = 0 Then
            Orange_Room11.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        Else
            Orange_Room12.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        End If
    ElseIf Direction = 3 Then
        If Correct1 = 2 Then
            Yellow_Room2.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        Else
            Orange_Room12.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        End If
    End If
    Timer1.Enabled = False
    ProgressBar1.Visible = False
    Label2.Visible = False
    Progress = 0
    Button1.Enabled = True
    Button2.Enabled = True
    Button3.Enabled = True
    If FeedbackGiven1 = True Then
        Label5.Visible = True
    End If
    If FeedbackGiven2 = True Then
        Label6.Visible = True
    End If
    If FeedbackGiven3 = True Then
        Label7.Visible = True
    End If
  End If
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer2.Tick
  Timer2.Enabled = False
  If FeedbackGiven1 = True Then
    Label5.Visible = True
  End If
  If FeedbackGiven2 = True Then
    Label6.Visible = True
  End If
  If FeedbackGiven3 = True Then
    Label7.Visible = True
  End If
```

```
        Button1.Enabled = True
        Button2.Enabled = True
        Button3.Enabled = True
    End Sub

End Class
```

B.10 Orange_Room11.vb (O_R12.vb, O_R21.vb, O_R22.vb , O_R31.vb , and O_R32.vb essentially the same)

```
Public Class Orange_Room11

    Dim Period As Integer
    Dim Room As Single
    Dim Maze As Integer
    Dim Direction As Integer
    Dim Progress As Integer
    Dim IntMillisecond As Integer
    Dim Int1 As Integer
    Dim Int2 As Integer
    Dim Int3 As Integer
    Dim Int4 As Integer
    Dim Int5 As Integer
    Dim Int6 As Integer
    Dim Int7 As Integer
    Dim Int8 As Integer
    Dim Int9 As Integer
    Public FeedbackGiven As Boolean = False

    Private Sub Orange_Room11_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Period = ID.Period
        Maze = ID.Maze
        Room = 1.1
        CountForm(Me)
        If Period < 2 Then
            If ID.Feedback <= 1 Then
                Label1.Visible = True
            End If
        End If

        IntMillisecond = Date.Now.Millisecond

        'Determine images for PictureBoxes
        Int1 = CInt(Rnd() * IntMillisecond)
        Int2 = CInt(Rnd() * IntMillisecond)
        Int3 = CInt(Rnd() * IntMillisecond)
        Int4 = CInt(Rnd() * IntMillisecond)
        Int5 = CInt(Rnd() * IntMillisecond)
        Int6 = CInt(Rnd() * IntMillisecond)
        Int7 = CInt(Rnd() * IntMillisecond)
        Int8 = CInt(Rnd() * IntMillisecond)
        Int9 = CInt(Rnd() * IntMillisecond)

        If ID.Cue = 1 Then
            "Color of maze is put in place
            If ID.Color1 = 1 Then
                PictureBox13.Image = My.Resources.Room1
            ElseIf ID.Color1 = 2 Then
                PictureBox13.Image = My.Resources.Room2
            Else
                PictureBox13.Image = My.Resources.Room3
            End If
```

```
"Ensures non-overlapping of images for corners and wall images
If Int6 Mod 3 <> 0 Then
    Int7 = 3
End If
If Int8 Mod 3 <> 0 Then
    Int5 = 3
End If
'Presence/absence of portrait is the cue
If Int1 Mod 3 = 1 Then
    PictureBox1.Image = My.Resources.Torch1
    PictureBox1.Visible = True
ElseIf Int1 Mod 3 = 2 Then
    PictureBox1.Image = My.Resources.Torch3
    PictureBox1.Visible = True
End If
If Int2 Mod 3 = 1 Then
    PictureBox2.Image = My.Resources.Torch
    PictureBox2.Visible = True
ElseIf Int2 Mod 3 = 2 Then
    PictureBox2.Image = My.Resources.Torch2
    PictureBox2.Visible = True
End If
If Int3 Mod 8 = 0 Then
    PictureBox3.Image = My.Resources.deer_head
    PictureBox3.Visible = True
ElseIf Int3 Mod 8 = 1 Then
    PictureBox3.Image = My.Resources.Picture2
    PictureBox3.Width = 129
    PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 8 = 2 Then
    PictureBox3.Image = My.Resources.Picture3
    PictureBox3.Width = 108
    PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 8 = 3 Then
Else
    PictureBox3.Image = My.Resources.Portrait
    PictureBox3.Visible = True
End If
If Int3 Mod 8 > 3 Then
    If Int4 Mod 4 = 1 Then
        PictureBox4.Image = My.Resources.deer_head
        PictureBox4.Visible = True
    ElseIf Int4 Mod 4 = 2 Then
        PictureBox4.Image = My.Resources.Picture2
        PictureBox4.Width = 129
        PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
        PictureBox4.Visible = True
    ElseIf Int4 Mod 4 = 3 Then
        PictureBox4.Image = My.Resources.Picture3
        PictureBox4.Width = 108
        PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
        PictureBox4.Visible = True
    End If
Else
    PictureBox4.Image = My.Resources.Portrait
    PictureBox4.Visible = True
End If
If Int5 Mod 3 = 1 Then
    PictureBox5.Image = My.Resources.Clock2
    PictureBox5.Height = 200
    PictureBox5.Width = 59
```

```
            PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
            PictureBox5.Visible = True
        ElseIf Int5 Mod 3 = 2 Then
            PictureBox5.Image = My.Resources.masters_chair
            PictureBox5.Height = 200
            PictureBox5.Width = 132
            PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
            PictureBox5.Visible = True
        End If
        If Int6 Mod 3 = 1 Then
            PictureBox6.Image = My.Resources.Clock1
            PictureBox6.Height = 200
            PictureBox6.Width = 59
            PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        ElseIf Int6 Mod 3 = 2 Then
            PictureBox6.Image = My.Resources.masters_chair1
            PictureBox6.Height = 200
            PictureBox6.Width = 132
            PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        End If
        If Int7 Mod 3 = 1 Then
            PictureBox7.Image = My.Resources.Secretaire
            PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
            PictureBox7.Visible = True
        ElseIf Int7 Mod 3 = 2 Then
            PictureBox7.Image = My.Resources.Fireplace
            PictureBox7.Height = 133
            PictureBox7.Width = 153
            PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
            PictureBox7.Visible = True
        End If
        If Int8 Mod 3 = 1 Then
            PictureBox8.Image = My.Resources.Secretaire1
            PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
            PictureBox8.Visible = True
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    ElseIf ID.Cue = 2 Then
        "Color of maze is put in place
        If ID.Color2 = 1 Then
            PictureBox13.Image = My.Resources.Room1
        ElseIf ID.Color2 = 2 Then
            PictureBox13.Image = My.Resources.Room2
        Else
            PictureBox13.Image = My.Resources.Room3
        End If
        "Ensures non-overlapping of images for corners and wall images
        If Int6 Mod 3 <> 0 Then
            Int7 = 3
```

```
End If
If Int8 Mod 3 <> 0 Then
   Int5 = 3
End If
'Type of torch is the cue
If Int1 Mod 4 = 1 Then
   PictureBox1.Image = My.Resources.Torch1
   PictureBox1.Visible = True
ElseIf Int1 Mod 4 > 1 Then
   PictureBox1.Image = My.Resources.Torch3
   PictureBox1.Visible = True
End If
If Int1 Mod 4 > 1 Then
   If Int2 Mod 4 = 1 Then
      PictureBox2.Image = My.Resources.Torch
      PictureBox2.Visible = True
   ElseIf Int2 Mod 4 >= 2 Then
      PictureBox2.Image = My.Resources.Torch2
      PictureBox2.Visible = True
   End If
Else
   PictureBox2.Image = My.Resources.Torch2
   PictureBox2.Visible = True
End If
If Int3 Mod 5 = 1 Then
   PictureBox3.Image = My.Resources.deer_head
   PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 2 Then
   PictureBox3.Image = My.Resources.Picture2
   PictureBox3.Width = 129
   PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
   PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 3 Then
   PictureBox3.Image = My.Resources.Picture3
   PictureBox3.Width = 108
   PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
   PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 4 Then
   PictureBox3.Image = My.Resources.Portrait
   PictureBox3.Visible = True
End If
If Int4 Mod 5 = 1 Then
   PictureBox4.Image = My.Resources.deer_head
   PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 2 Then
   PictureBox4.Image = My.Resources.Picture2
   PictureBox4.Width = 129
   PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
   PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 3 Then
   PictureBox4.Image = My.Resources.Picture3
   PictureBox4.Width = 108
   PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
   PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 4 Then
   PictureBox4.Image = My.Resources.Portrait
   PictureBox4.Visible = True
End If
If Int5 Mod 3 = 1 Then
   PictureBox5.Image = My.Resources.Clock2
   PictureBox5.Height = 200
   PictureBox5.Width = 59
   PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
```

```
            PictureBox5.Visible = True
        ElseIf Int5 Mod 3 = 2 Then
            PictureBox5.Image = My.Resources.masters_chair
            PictureBox5.Height = 200
            PictureBox5.Width = 132
            PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
            PictureBox5.Visible = True
        End If
        If Int6 Mod 3 = 1 Then
            PictureBox6.Image = My.Resources.Clock1
            PictureBox6.Height = 200
            PictureBox6.Width = 59
            PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        ElseIf Int6 Mod 3 = 2 Then
            PictureBox6.Image = My.Resources.masters_chair1
            PictureBox6.Height = 200
            PictureBox6.Width = 132
            PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        End If
        If Int7 Mod 3 = 1 Then
            PictureBox7.Image = My.Resources.Secretaire
            PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
            PictureBox7.Visible = True
        ElseIf Int7 Mod 3 = 2 Then
            PictureBox7.Image = My.Resources.Fireplace
            PictureBox7.Height = 133
            PictureBox7.Width = 153
            PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
            PictureBox7.Visible = True
        End If
        If Int8 Mod 3 = 1 Then
            PictureBox8.Image = My.Resources.Secretaire1
            PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
            PictureBox8.Visible = True
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    Else
        "Color of maze is put in place
        If ID.Color3 = 1 Then
            PictureBox13.Image = My.Resources.Room1
        ElseIf ID.Color3 = 2 Then
            PictureBox13.Image = My.Resources.Room2
        Else
            PictureBox13.Image = My.Resources.Room3
        End If
        "Ensures non-overlapping of images for corners and wall images
        If Int7 Mod 3 <> 0 Then            'If PB7 shows an image then
            Int6 = 4                    'Make PB6 be blank
        End If
```

```
If Int8 Mod 3 <> 0 Then            'If PB8 shows an image then
   Int5 = 4                    'Make PB5 be blank
End If
"Make sure at least one chair shows up
If Int5 Mod 4 < 2 Then         'If PB5 is not a chair then
   If Int6 Mod 4 < 2 Then      'If PB6 is not a chair then
      If Int1 Mod 2 = 1 Then  'Picks a head or tails
         Int5 = 6           'Make PB5 be a chair
         Int8 = 3           'Make PB8 be blank
      Else
         Int6 = 6           'Make PB6 be a chair
         Int7 = 3           'Make PB7 be blank
      End If
   End If
End If
'Presence/absence of chair(s) is cue
If Int1 Mod 3 = 1 Then
   PictureBox1.Image = My.Resources.Torch1
   PictureBox1.Visible = True
ElseIf Int1 Mod 3 = 2 Then
   PictureBox1.Image = My.Resources.Torch3
   PictureBox1.Visible = True
End If
If Int2 Mod 3 = 1 Then
   PictureBox2.Image = My.Resources.Torch
   PictureBox2.Visible = True
ElseIf Int2 Mod 3 = 2 Then
   PictureBox2.Image = My.Resources.Torch2
   PictureBox2.Visible = True
End If
If Int3 Mod 5 = 1 Then
   PictureBox3.Image = My.Resources.deer_head
   PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 2 Then
   PictureBox3.Image = My.Resources.Picture2
   PictureBox3.Width = 129
   PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
   PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 3 Then
   PictureBox3.Image = My.Resources.Picture3
   PictureBox3.Width = 108
   PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
   PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 4 Then
   PictureBox3.Image = My.Resources.Portrait
   PictureBox3.Visible = True
End If
If Int4 Mod 5 = 1 Then
   PictureBox4.Image = My.Resources.deer_head
   PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 2 Then
   PictureBox4.Image = My.Resources.Picture2
   PictureBox4.Width = 129
   PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
   PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 3 Then
   PictureBox4.Image = My.Resources.Picture3
   PictureBox4.Width = 108
   PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
   PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 4 Then
   PictureBox4.Image = My.Resources.Portrait
   PictureBox4.Visible = True
```

```
        End If
        If Int5 Mod 4 = 1 Then
            PictureBox5.Image = My.Resources.Clock2
            PictureBox5.Height = 200
            PictureBox5.Width = 59
            PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
            PictureBox5.Visible = True
        ElseIf Int5 Mod 4 >= 2 Then
            PictureBox5.Image = My.Resources.masters_chair
            PictureBox5.Height = 200
            PictureBox5.Width = 132
            PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
            PictureBox5.Visible = True
        End If
        If Int6 Mod 4 = 1 Then
            PictureBox6.Image = My.Resources.Clock1
            PictureBox6.Height = 200
            PictureBox6.Width = 59
            PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        ElseIf Int6 Mod 4 >= 2 Then
            PictureBox6.Image = My.Resources.masters_chair1
            PictureBox6.Height = 200
            PictureBox6.Width = 132
            PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        End If
        If Int7 Mod 3 = 1 Then
            PictureBox7.Image = My.Resources.Secretaire
            PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
            PictureBox7.Visible = True
        ElseIf Int7 Mod 3 = 2 Then
            PictureBox7.Image = My.Resources.Fireplace
            PictureBox7.Height = 133
            PictureBox7.Width = 153
            PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
            PictureBox7.Visible = True
        End If
        If Int8 Mod 3 = 1 Then
            PictureBox8.Image = My.Resources.Secretaire1
            PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
            PictureBox8.Visible = True
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    End If

    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & vbCrLf, True)
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
        Direction = 1
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Timer1.Enabled = True
        If Period = 1 Then
            If ID.Feedback <= 2 Then
                If Yellow_Room1.Correct1 = 0 Then
                    Yellow_Room1.FeedbackGiven2 = True
                ElseIf Yellow_Room1.Correct1 > 0 Then
                    Yellow_Room1.FeedbackGiven1 = True
                End If
            End If
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Direction = 2
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Timer1.Enabled = True
        If Period = 1 Then
            If ID.Feedback <= 2 Then
                If Yellow_Room1.Correct1 = 0 Then
                    Yellow_Room1.FeedbackGiven2 = True
                ElseIf Yellow_Room1.Correct1 > 0 Then
                    Yellow_Room1.FeedbackGiven1 = True
                End If
            End If
        End If
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Direction = 3
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Timer1.Enabled = True
        If Period = 1 Then
            If ID.Feedback <= 2 Then
                If Yellow_Room1.Correct1 = 0 Then
                    Yellow_Room1.FeedbackGiven2 = True
                ElseIf Yellow_Room1.Correct1 > 0 Then
                    Yellow_Room1.FeedbackGiven1 = True
```

```
            End If
        End If
    End If
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Direction = 4
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
    ProgressBar1.Visible = True
    Label2.Visible = True
    Label2.Text = "Proceeding backward..."
    Button1.Enabled = False
    Button2.Enabled = False
    Button3.Enabled = False
    Button4.Enabled = False
    Timer2.Enabled = True
End Sub

'MAKES TIMER COUNT DOWN
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Value = Progress
    If Progress < 100 Then
        Progress = Progress + Yellow_Room1.AddNumber
    ElseIf Progress = 100 Then
        If Direction = 1 Then
            Red_Room111.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        ElseIf Direction = 2 Then
            Red_Room112.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        ElseIf Direction = 3 Then
            Red_Room113.Show()
            Me.Hide()
            ProgressBar1.Value = 0
        End If
        Timer1.Enabled = False
        ProgressBar1.Value = 0
        ProgressBar1.Visible = False
        Label2.Visible = False
        Progress = 0
        Button1.Enabled = True
        Button2.Enabled = True
        Button3.Enabled = True
        Button4.Enabled = True
        If Yellow_Room1.FeedbackGiven1 = True Then
            Yellow_Room1.Label5.Visible = True
        End If
        If Yellow_Room1.FeedbackGiven2 = True Then
            Yellow_Room1.Label6.Visible = True
        End If
        If Yellow_Room1.FeedbackGiven3 = True Then
            Yellow_Room1.Label7.Visible = True
        End If
    End If
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer2.Tick
    ProgressBar1.Value = Progress
    If Progress < 100 Then
```

```
            Progress = Progress + Yellow_Room1.AddNumber
        ElseIf Progress = 100 Then
            If Yellow_Room1.FeedbackGiven1 = True Then
                Yellow_Room1.Label5.Visible = True
            End If
            If Yellow_Room1.FeedbackGiven2 = True Then
                Yellow_Room1.Label6.Visible = True
            End If
            If Yellow_Room1.FeedbackGiven3 = True Then
                Yellow_Room1.Label7.Visible = True
            End If
            Yellow_Room1.Show()
            Me.Hide()
            Timer2.Enabled = False
            ProgressBar1.Value = 0
            ProgressBar1.Visible = False
            Label2.Visible = False
            Label2.Text = "Proceeding forward..."
            Progress = 0
            Button1.Enabled = True
            Button2.Enabled = True
            Button3.Enabled = True
            Button4.Enabled = True
        End If
    End Sub

    Private Sub Orange_Room11_Activated(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Activated
        'To show feedback after delay to red room
        If FeedbackGiven = True Then
            Label1.Visible = True
        Else
            Label1.Visible = False
        End If
    End Sub
End Class
```

B.11 Red_Room111.vb (R_R112.vb, R_R113.vb, R_R121.vb, R_R122.vb, R_R123.vb, R_R211.vb, R_R212.vb, R_R213.vb, R_R221.vb, R_R222.vb, R_R223.vb, R_R311.vb, R_R312.vb, R_R313.vb, R_R321.vb, R_R322.vb, and R_R323.vb essentially the same)

```
Public Class Red_Room111

    Dim Period As Integer
    Dim Room As Single
    Dim Maze As Integer
    Dim Direction As Integer
    Dim Progress As Integer
    Dim IntMillisecond As Integer
    Dim Int1 As Integer
    Dim Int2 As Integer
    Dim Int3 As Integer
    Dim Int4 As Integer
    Dim Int5 As Integer
    Dim Int6 As Integer
    Dim Int7 As Integer
    Dim Int8 As Integer
    Dim Int9 As Integer
    Public FeedbackGiven As Boolean = False

    Private Sub Red_Room111_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
```

```
Period = ID.Period
Maze = ID.Maze
Room = 1.11
CountForm(Me)
If Period < 2 Then
   If ID.Feedback <= 2 Then
      Label1.Visible = True
      FeedbackGiven = True
      'Flags orange rooms to show feedback now
      Orange_Room11.FeedbackGiven = True
   End If
End If

IntMillisecond = Date.Now.Millisecond

'Determine images for PictureBoxes
Int1 = CInt(Rnd() * IntMillisecond)
Int2 = CInt(Rnd() * IntMillisecond)
Int3 = CInt(Rnd() * IntMillisecond)
Int4 = CInt(Rnd() * IntMillisecond)
Int5 = CInt(Rnd() * IntMillisecond)
Int6 = CInt(Rnd() * IntMillisecond)
Int7 = CInt(Rnd() * IntMillisecond)
Int8 = CInt(Rnd() * IntMillisecond)
Int9 = CInt(Rnd() * IntMillisecond)

''Obfuscate learning in second period
'If Period = 1 Then
'   If ID.Cue = 1 Then
'      ''Color of maze is put in place
'      If ID.Color1 = 1 Then
'         PictureBox13.Image = My.Resources.Room1
'      ElseIf ID.Color1 = 2 Then
'         PictureBox13.Image = My.Resources.Room2
'      Else
'         PictureBox13.Image = My.Resources.Room3
'      End If
'      'Presence/absence of portrait is the cue
'      If Int3 Mod 2 = 1 Then
'         PictureBox3.Image = My.Resources.Portrait
'         PictureBox3.Visible = True
'      End If
'      If Int3 Mod 2 <> 1 Then
'         PictureBox4.Image = My.Resources.Portrait
'         PictureBox4.Visible = True
'      End If
'   ElseIf ID.Cue = 2 Then
'      ''Color of maze is put in place
'      If ID.Color2 = 1 Then
'         PictureBox13.Image = My.Resources.Room1
'      ElseIf ID.Color2 = 2 Then
'         PictureBox13.Image = My.Resources.Room2
'      Else
'         PictureBox13.Image = My.Resources.Room3
'      End If
'      'Type of torch is the cue
'      If Int1 Mod 2 = 1 Then
'         PictureBox1.Image = My.Resources.Torch3
'         PictureBox1.Visible = True
'      End If
'      If Int1 Mod 2 = 1 Then
'         If Int2 Mod 2 = 1 Then
'            PictureBox2.Image = My.Resources.Torch2
```

```
'            PictureBox2.Visible = True
'        End If
'    Else
'        PictureBox2.Image = My.Resources.Torch2
'        PictureBox2.Visible = True
'    End If
'  Else
'    "Color of maze is put in place
'    If ID.Color3 = 1 Then
'        PictureBox13.Image = My.Resources.Room1
'    ElseIf ID.Color3 = 2 Then
'        PictureBox13.Image = My.Resources.Room2
'    Else
'        PictureBox13.Image = My.Resources.Room3
'    End If
'    'Presence/absence of chair(s) is cue
'    If Int5 Mod 2 = 1 Then
'        PictureBox5.Image = My.Resources.masters_chair
'        PictureBox5.Height = 200
'        PictureBox5.Width = 132
'        PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
'        PictureBox5.Visible = True
'    End If
'    "Make sure at least one chair shows up
'    If Int5 Mod 2 <> 1 Then
'        Int6 = 3
'    End If
'    If Int6 Mod 2 = 1 Then
'        PictureBox6.Image = My.Resources.masters_chair1
'        PictureBox6.Height = 200
'        PictureBox6.Width = 132
'        PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
'        PictureBox6.Visible = True
'    End If
'  End If
'ElseIf Period = 2 Then
If ID.Cue = 1 Then
  "Color of maze is put in place
  If ID.Color1 = 1 Then
    PictureBox13.Image = My.Resources.Room1
  ElseIf ID.Color1 = 2 Then
    PictureBox13.Image = My.Resources.Room2
  Else
    PictureBox13.Image = My.Resources.Room3
  End If
  "Ensures non-overlapping of images for corners and wall images
  If Int6 Mod 3 <> 0 Then
    Int7 = 3
  End If
  If Int8 Mod 3 <> 0 Then
    Int5 = 3
  End If
  'Presence/absence of portrait is the cue
  If Int1 Mod 3 = 1 Then
    PictureBox1.Image = My.Resources.Torch1
    PictureBox1.Visible = True
  ElseIf Int1 Mod 3 = 2 Then
    PictureBox1.Image = My.Resources.Torch3
    PictureBox1.Visible = True
  End If
  If Int2 Mod 3 = 1 Then
    PictureBox2.Image = My.Resources.Torch
    PictureBox2.Visible = True
```

```
ElseIf Int2 Mod 3 = 2 Then
    PictureBox2.Image = My.Resources.Torch2
    PictureBox2.Visible = True
End If
If Int3 Mod 8 = 0 Then
    PictureBox3.Image = My.Resources.deer_head
    PictureBox3.Visible = True
ElseIf Int3 Mod 8 = 1 Then
    PictureBox3.Image = My.Resources.Picture2
    PictureBox3.Width = 129
    PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 8 = 2 Then
    PictureBox3.Image = My.Resources.Picture3
    PictureBox3.Width = 108
    PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 8 = 3 Then
Else
    PictureBox3.Image = My.Resources.Portrait
    PictureBox3.Visible = True
End If
If Int3 Mod 8 > 3 Then
    If Int4 Mod 4 = 1 Then
        PictureBox4.Image = My.Resources.deer_head
        PictureBox4.Visible = True
    ElseIf Int4 Mod 4 = 2 Then
        PictureBox4.Image = My.Resources.Picture2
        PictureBox4.Width = 129
        PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
        PictureBox4.Visible = True
    ElseIf Int4 Mod 4 = 3 Then
        PictureBox4.Image = My.Resources.Picture3
        PictureBox4.Width = 108
        PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
        PictureBox4.Visible = True
    End If
Else
    PictureBox4.Image = My.Resources.Portrait
    PictureBox4.Visible = True
End If
If Int5 Mod 3 = 1 Then
    PictureBox5.Image = My.Resources.Clock2
    PictureBox5.Height = 200
    PictureBox5.Width = 59
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
ElseIf Int5 Mod 3 = 2 Then
    PictureBox5.Image = My.Resources.masters_chair
    PictureBox5.Height = 200
    PictureBox5.Width = 132
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
End If
If Int6 Mod 3 = 1 Then
    PictureBox6.Image = My.Resources.Clock1
    PictureBox6.Height = 200
    PictureBox6.Width = 59
    PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
    PictureBox6.Visible = True
ElseIf Int6 Mod 3 = 2 Then
    PictureBox6.Image = My.Resources.masters_chair1
    PictureBox6.Height = 200
```

```
            PictureBox6.Width = 132
            PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        End If
        If Int7 Mod 3 = 1 Then
            PictureBox7.Image = My.Resources.Secretaire
            PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
            PictureBox7.Visible = True
        ElseIf Int7 Mod 3 = 2 Then
            PictureBox7.Image = My.Resources.Fireplace
            PictureBox7.Height = 133
            PictureBox7.Width = 153
            PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
            PictureBox7.Visible = True
        End If
        If Int8 Mod 3 = 1 Then
            PictureBox8.Image = My.Resources.Secretaire1
            PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
            PictureBox8.Visible = True
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    ElseIf ID.Cue = 2 Then
        "Color of maze is put in place
        If ID.Color2 = 1 Then
            PictureBox13.Image = My.Resources.Room1
        ElseIf ID.Color2 = 2 Then
            PictureBox13.Image = My.Resources.Room2
        Else
            PictureBox13.Image = My.Resources.Room3
        End If
        "Ensures non-overlapping of images for corners and wall images
        If Int6 Mod 3 <> 0 Then
            Int7 = 3
        End If
        If Int8 Mod 3 <> 0 Then
            Int5 = 3
        End If
        'Type of torch is the cue
        If Int1 Mod 4 = 1 Then
            PictureBox1.Image = My.Resources.Torch1
            PictureBox1.Visible = True
        ElseIf Int1 Mod 4 > 1 Then
            PictureBox1.Image = My.Resources.Torch3
            PictureBox1.Visible = True
        End If
        If Int1 Mod 4 > 1 Then
            If Int2 Mod 4 = 1 Then
                PictureBox2.Image = My.Resources.Torch
                PictureBox2.Visible = True
            ElseIf Int2 Mod 4 >= 2 Then
                PictureBox2.Image = My.Resources.Torch2
```

```
        PictureBox2.Visible = True
    End If
Else
    PictureBox2.Image = My.Resources.Torch2
    PictureBox2.Visible = True
End If
If Int3 Mod 5 = 1 Then
    PictureBox3.Image = My.Resources.deer_head
    PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 2 Then
    PictureBox3.Image = My.Resources.Picture2
    PictureBox3.Width = 129
    PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 3 Then
    PictureBox3.Image = My.Resources.Picture3
    PictureBox3.Width = 108
    PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 4 Then
    PictureBox3.Image = My.Resources.Portrait
    PictureBox3.Visible = True
End If
If Int4 Mod 5 = 1 Then
    PictureBox4.Image = My.Resources.deer_head
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 2 Then
    PictureBox4.Image = My.Resources.Picture2
    PictureBox4.Width = 129
    PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 3 Then
    PictureBox4.Image = My.Resources.Picture3
    PictureBox4.Width = 108
    PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 4 Then
    PictureBox4.Image = My.Resources.Portrait
    PictureBox4.Visible = True
End If
If Int5 Mod 3 = 1 Then
    PictureBox5.Image = My.Resources.Clock2
    PictureBox5.Height = 200
    PictureBox5.Width = 59
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
ElseIf Int5 Mod 3 = 2 Then
    PictureBox5.Image = My.Resources.masters_chair
    PictureBox5.Height = 200
    PictureBox5.Width = 132
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
End If
If Int6 Mod 3 = 1 Then
    PictureBox6.Image = My.Resources.Clock1
    PictureBox6.Height = 200
    PictureBox6.Width = 59
    PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
    PictureBox6.Visible = True
ElseIf Int6 Mod 3 = 2 Then
    PictureBox6.Image = My.Resources.masters_chair1
    PictureBox6.Height = 200
    PictureBox6.Width = 132
```

```
        PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
        PictureBox6.Visible = True
    End If
    If Int7 Mod 3 = 1 Then
        PictureBox7.Image = My.Resources.Secretaire
        PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
        PictureBox7.Visible = True
    ElseIf Int7 Mod 3 = 2 Then
        PictureBox7.Image = My.Resources.Fireplace
        PictureBox7.Height = 133
        PictureBox7.Width = 153
        PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
        PictureBox7.Visible = True
    End If
    If Int8 Mod 3 = 1 Then
        PictureBox8.Image = My.Resources.Secretaire1
        PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
        PictureBox8.Visible = True
    ElseIf Int8 Mod 3 = 2 Then
        PictureBox8.Image = My.Resources.Fireplace
        PictureBox8.Height = 133
        PictureBox8.Width = 153
        PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
        PictureBox8.Visible = True
    End If
    If Int9 Mod 3 = 1 Then
        PictureBox9.Image = My.Resources.rug1
        PictureBox9.Visible = True
    ElseIf Int9 Mod 3 = 2 Then
        PictureBox9.Image = My.Resources.rug2
        PictureBox9.Visible = True
    End If
Else
    "Color of maze is put in place
    If ID.Color3 = 1 Then
        PictureBox13.Image = My.Resources.Room1
    ElseIf ID.Color3 = 2 Then
        PictureBox13.Image = My.Resources.Room2
    Else
        PictureBox13.Image = My.Resources.Room3
    End If
    "Ensures non-overlapping of images for corners and wall images
    If Int7 Mod 3 <> 0 Then           'If PB7 shows an image then
        Int6 = 4                      'Make PB6 be blank
    End If
    If Int8 Mod 3 <> 0 Then           'If PB8 shows an image then
        Int5 = 4                      'Make PB5 be blank
    End If
    "Make sure at least one chair shows up
    If Int5 Mod 4 < 2 Then            'If PB5 is not a chair then
        If Int6 Mod 4 < 2 Then        'If PB6 is not a chair then
            If Int1 Mod 2 = 1 Then    'Picks a head or tails
                Int5 = 6              'Make PB5 be a chair
                Int8 = 3              'Make PB8 be blank
            Else
                Int6 = 6              'Make PB6 be a chair
                Int7 = 3              'Make PB7 be blank
            End If
        End If
    End If
    'Presence/absence of chair(s) is cue
    If Int1 Mod 3 = 1 Then
        PictureBox1.Image = My.Resources.Torch1
```

```
    PictureBox1.Visible = True
ElseIf Int1 Mod 3 = 2 Then
    PictureBox1.Image = My.Resources.Torch3
    PictureBox1.Visible = True
End If
If Int2 Mod 3 = 1 Then
    PictureBox2.Image = My.Resources.Torch
    PictureBox2.Visible = True
ElseIf Int2 Mod 3 = 2 Then
    PictureBox2.Image = My.Resources.Torch2
    PictureBox2.Visible = True
End If
If Int3 Mod 5 = 1 Then
    PictureBox3.Image = My.Resources.deer_head
    PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 2 Then
    PictureBox3.Image = My.Resources.Picture2
    PictureBox3.Width = 129
    PictureBox3.Location = New Point(PictureBox3.Location.X - 17, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 3 Then
    PictureBox3.Image = My.Resources.Picture3
    PictureBox3.Width = 108
    PictureBox3.Location = New Point(PictureBox3.Location.X - 7, PictureBox3.Location.Y)
    PictureBox3.Visible = True
ElseIf Int3 Mod 5 = 4 Then
    PictureBox3.Image = My.Resources.Portrait
    PictureBox3.Visible = True
End If
If Int4 Mod 5 = 1 Then
    PictureBox4.Image = My.Resources.deer_head
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 2 Then
    PictureBox4.Image = My.Resources.Picture2
    PictureBox4.Width = 129
    PictureBox4.Location = New Point(PictureBox4.Location.X - 17, PictureBox4.Location.Y)
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 3 Then
    PictureBox4.Image = My.Resources.Picture3
    PictureBox4.Width = 108
    PictureBox4.Location = New Point(PictureBox4.Location.X - 7, PictureBox4.Location.Y)
    PictureBox4.Visible = True
ElseIf Int4 Mod 5 = 4 Then
    PictureBox4.Image = My.Resources.Portrait
    PictureBox4.Visible = True
End If
If Int5 Mod 4 = 1 Then
    PictureBox5.Image = My.Resources.Clock2
    PictureBox5.Height = 200
    PictureBox5.Width = 59
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
ElseIf Int5 Mod 4 >= 2 Then
    PictureBox5.Image = My.Resources.masters_chair
    PictureBox5.Height = 200
    PictureBox5.Width = 132
    PictureBox5.Location = New Point(PictureBox5.Location.X - 8, PictureBox5.Location.Y)
    PictureBox5.Visible = True
End If
If Int6 Mod 4 = 1 Then
    PictureBox6.Image = My.Resources.Clock1
    PictureBox6.Height = 200
    PictureBox6.Width = 59
```

```
            PictureBox6.Location = New Point(700, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        ElseIf Int6 Mod 4 >= 2 Then
            PictureBox6.Image = My.Resources.masters_chair1
            PictureBox6.Height = 200
            PictureBox6.Width = 132
            PictureBox6.Location = New Point(PictureBox6.Location.X - 37, PictureBox6.Location.Y)
            PictureBox6.Visible = True
        End If
        If Int7 Mod 3 = 1 Then
            PictureBox7.Image = My.Resources.Secretaire
            PictureBox7.Location = New Point(PictureBox7.Location.X, PictureBox7.Location.Y + 36)
            PictureBox7.Visible = True
        ElseIf Int7 Mod 3 = 2 Then
            PictureBox7.Image = My.Resources.Fireplace
            PictureBox7.Height = 133
            PictureBox7.Width = 153
            PictureBox7.Location = New Point(PictureBox7.Location.X - 10, PictureBox7.Location.Y + 17)
            PictureBox7.Visible = True
        End If
        If Int8 Mod 3 = 1 Then
            PictureBox8.Image = My.Resources.Secretaire1
            PictureBox8.Location = New Point(PictureBox8.Location.X, PictureBox8.Location.Y + 36)
            PictureBox8.Visible = True
        ElseIf Int8 Mod 3 = 2 Then
            PictureBox8.Image = My.Resources.Fireplace
            PictureBox8.Height = 133
            PictureBox8.Width = 153
            PictureBox8.Location = New Point(PictureBox8.Location.X - 10, PictureBox8.Location.Y + 17)
            PictureBox8.Visible = True
        End If
        If Int9 Mod 3 = 1 Then
            PictureBox9.Image = My.Resources.rug1
            PictureBox9.Visible = True
        ElseIf Int9 Mod 3 = 2 Then
            PictureBox9.Image = My.Resources.rug2
            PictureBox9.Visible = True
        End If
    End If
    'End If

    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & vbCrLf, True)
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Direction = 1
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
    ProgressBar1.Visible = True
    Label2.Visible = True
    Button1.Enabled = False
    Button2.Enabled = False
    Button3.Enabled = False
    Button4.Enabled = False
    Timer1.Enabled = True
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Direction = 2
```

```
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Timer1.Enabled = True
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Direction = 3
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        ProgressBar1.Visible = True
        Label2.Visible = True
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Timer1.Enabled = True
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        Direction = 4
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        ProgressBar1.Visible = True
        Label2.Visible = True
        Label2.Text = "Proceeding backward..."
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Timer2.Enabled = True
    End Sub

    'MAKES TIMER COUNT DOWN
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
        ProgressBar1.Value = Progress
        If Progress < 100 Then
            Progress = Progress + Yellow_Room1.AddNumber
        ElseIf Progress = 100 Then
            DarkRed_Room.Show()
            Me.Hide()
            If Direction = 1 Then
                DarkRed_Room.Room = 1.111
            ElseIf Direction = 2 Then
                DarkRed_Room.Room = 1.112
            ElseIf Direction = 3 Then
                DarkRed_Room.Room = 1.113
            End If
            Timer1.Enabled = False
            ProgressBar1.Value = 0
            ProgressBar1.Visible = False
            Label2.Visible = False
            Progress = 0
            Button1.Enabled = True
            Button2.Enabled = True
```

```
            Button3.Enabled = True
            Button4.Enabled = True
        End If
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer2.Tick
        ProgressBar1.Value = Progress
        If Progress < 100 Then
            Progress = Progress + Yellow_Room1.AddNumber
        ElseIf Progress = 100 Then
            Orange_Room11.Show()
            Me.Hide()
            Timer2.Enabled = False
            ProgressBar1.Value = 0
            ProgressBar1.Visible = False
            Label2.Visible = False
            Label2.Text = "Proceeding forward..."
            Progress = 0
            Button1.Enabled = True
            Button2.Enabled = True
            Button3.Enabled = True
            Button4.Enabled = True
        End If
    End Sub

    Private Sub Red_Room111_Activated(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Activated
        'To show feedback after delay to dark red room
        If FeedbackGiven = True Then
            Label1.Visible = True
        Else
            Label1.Visible = False
        End If
    End Sub

End Class
```

B.12 DarkRed_Room.vb

```
Public Class DarkRed_Room

    Dim Period As Integer
    Dim Maze As Integer
    Public Room As Single
    Dim Direction As Integer
    Dim Progress As Integer
    Dim PriorRoom As Single

    Private Sub DarkRed_Room_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Period = ID.Period
        Maze = ID.Maze
        CountForm(Me)
        If Period < 2 Then
            If ID.Feedback <= 3 Then
                Label1.Visible = True
            End If
        End If

        "Color of maze is put in place
        If ID.Cue = 1 Then
            If ID.Color1 = 1 Then
                PictureBox2.Image = My.Resources.DeadEnd1
            ElseIf ID.Color1 = 2 Then
```

```
            PictureBox2.Image = My.Resources.DeadEnd2
        Else
            PictureBox2.Image = My.Resources.DeadEnd3
        End If
    ElseIf ID.Cue = 2 Then
        If ID.Color2 = 1 Then
            PictureBox2.Image = My.Resources.DeadEnd1
        ElseIf ID.Color2 = 2 Then
            PictureBox2.Image = My.Resources.DeadEnd2
        Else
            PictureBox2.Image = My.Resources.DeadEnd3
        End If
    Else
        If ID.Color3 = 1 Then
            PictureBox2.Image = My.Resources.DeadEnd1
        ElseIf ID.Color3 = 2 Then
            PictureBox2.Image = My.Resources.DeadEnd2
        Else
            PictureBox2.Image = My.Resources.DeadEnd3
        End If
    End If
End Sub

Private Sub DarkRed_Room_Shown(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Shown
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & vbCrLf, True)
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Direction = 4
    My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
    ProgressBar1.Visible = True
    Label2.Visible = True
    Button4.Enabled = False
    Timer2.Enabled = True
End Sub

Sub ShowPriorRoom()
    PriorRoom = Int(Room * 100)
    If Period < 2 Then
        If ID.Feedback = 3 Then
            'Flags red, orange, and yellow rooms to show feedback now
            If PriorRoom = 111 Then
                If Yellow_Room1.Correct1 = 0 Then
                    Yellow_Room1.FeedbackGiven2 = True
                ElseIf Yellow_Room1.Correct1 > 0 Then
                    Yellow_Room1.FeedbackGiven1 = True
                End If
                Orange_Room11.FeedbackGiven = True
                Red_Room111.FeedbackGiven = True
                Red_Room111.Show()
            ElseIf PriorRoom = 112 Then
                If Yellow_Room1.Correct1 = 0 Then
                    Yellow_Room1.FeedbackGiven2 = True
                ElseIf Yellow_Room1.Correct1 > 0 Then
                    Yellow_Room1.FeedbackGiven1 = True
                End If
                Orange_Room11.FeedbackGiven = True
                Red_Room112.FeedbackGiven = True
                Red_Room112.Show()
            ElseIf PriorRoom = 113 Then
```

```
    If Yellow_Room1.Correct1 = 0 Then
        Yellow_Room1.FeedbackGiven2 = True
    ElseIf Yellow_Room1.Correct1 > 0 Then
        Yellow_Room1.FeedbackGiven1 = True
    End If
    Orange_Room11.FeedbackGiven = True
    Red_Room113.FeedbackGiven = True
    Red_Room113.Show()
ElseIf PriorRoom = 121 Then
    If Yellow_Room1.Correct1 < 2 Then
        Yellow_Room1.FeedbackGiven3 = True
    ElseIf Yellow_Room1.Correct1 = 2 Then
        Yellow_Room1.FeedbackGiven2 = True
    End If
    Orange_Room12.FeedbackGiven = True
    Red_Room121.FeedbackGiven = True
    Red_Room121.Show()
ElseIf PriorRoom = 122 Then
    If Yellow_Room1.Correct1 < 2 Then
        Yellow_Room1.FeedbackGiven3 = True
    ElseIf Yellow_Room1.Correct1 = 2 Then
        Yellow_Room1.FeedbackGiven2 = True
    End If
    Orange_Room12.FeedbackGiven = True
    Red_Room122.FeedbackGiven = True
    Red_Room122.Show()
ElseIf PriorRoom = 123 Then
    If Yellow_Room1.Correct1 < 2 Then
        Yellow_Room1.FeedbackGiven3 = True
    ElseIf Yellow_Room1.Correct1 = 2 Then
        Yellow_Room1.FeedbackGiven2 = True
    End If
    Orange_Room12.FeedbackGiven = True
    Red_Room123.FeedbackGiven = True
    Red_Room123.Show()
ElseIf PriorRoom = 211 Then
    If Yellow_Room2.Correct2 = 0 Then
        Yellow_Room2.FeedbackGiven2 = True
    ElseIf Yellow_Room2.Correct2 > 0 Then
        Yellow_Room2.FeedbackGiven1 = True
    End If
    Orange_Room21.FeedbackGiven = True
    Red_Room211.FeedbackGiven = True
    Red_Room211.Show()
ElseIf PriorRoom = 212 Then
    If Yellow_Room2.Correct2 = 0 Then
        Yellow_Room2.FeedbackGiven2 = True
    ElseIf Yellow_Room2.Correct2 > 0 Then
        Yellow_Room2.FeedbackGiven1 = True
    End If
    Orange_Room21.FeedbackGiven = True
    Red_Room212.FeedbackGiven = True
    Red_Room212.Show()
ElseIf PriorRoom = 213 Then
    If Yellow_Room2.Correct2 = 0 Then
        Yellow_Room2.FeedbackGiven2 = True
    ElseIf Yellow_Room2.Correct2 > 0 Then
        Yellow_Room2.FeedbackGiven1 = True
    End If
    Orange_Room21.FeedbackGiven = True
    Red_Room213.FeedbackGiven = True
    Red_Room213.Show()
ElseIf PriorRoom = 221 Then
```

```
    If Yellow_Room2.Correct2 < 2 Then
      Yellow_Room2.FeedbackGiven3 = True
    ElseIf Yellow_Room2.Correct2 = 2 Then
      Yellow_Room2.FeedbackGiven2 = True
    End If
    Orange_Room22.FeedbackGiven = True
    Red_Room221.FeedbackGiven = True
    Red_Room221.Show()
  ElseIf PriorRoom = 222 Then
    If Yellow_Room2.Correct2 < 2 Then
      Yellow_Room2.FeedbackGiven3 = True
    ElseIf Yellow_Room2.Correct2 = 2 Then
      Yellow_Room2.FeedbackGiven2 = True
    End If
    Orange_Room22.FeedbackGiven = True
    Red_Room222.FeedbackGiven = True
    Red_Room222.Show()
  ElseIf PriorRoom = 223 Then
    If Yellow_Room2.Correct2 < 2 Then
      Yellow_Room2.FeedbackGiven3 = True
    ElseIf Yellow_Room2.Correct2 = 2 Then
      Yellow_Room2.FeedbackGiven2 = True
    End If
    Orange_Room22.FeedbackGiven = True
    Red_Room223.FeedbackGiven = True
    Red_Room223.Show()
  ElseIf PriorRoom = 311 Then
    If Yellow_Room3.Correct3 = 0 Then
      Yellow_Room3.FeedbackGiven2 = True
    ElseIf Yellow_Room3.Correct3 > 0 Then
      Yellow_Room3.FeedbackGiven1 = True
    End If
    Orange_Room31.FeedbackGiven = True
    Red_Room311.FeedbackGiven = True
    Red_Room311.Show()
  ElseIf PriorRoom = 312 Then
    If Yellow_Room3.Correct3 = 0 Then
      Yellow_Room3.FeedbackGiven2 = True
    ElseIf Yellow_Room3.Correct3 > 0 Then
      Yellow_Room3.FeedbackGiven1 = True
    End If
    Orange_Room31.FeedbackGiven = True
    Red_Room312.FeedbackGiven = True
    Red_Room312.Show()
  ElseIf PriorRoom = 313 Then
    If Yellow_Room3.Correct3 = 0 Then
      Yellow_Room3.FeedbackGiven2 = True
    ElseIf Yellow_Room3.Correct3 > 0 Then
      Yellow_Room3.FeedbackGiven1 = True
    End If
    Orange_Room31.FeedbackGiven = True
    Red_Room313.FeedbackGiven = True
    Red_Room313.Show()
  ElseIf PriorRoom = 321 Then
    If Yellow_Room3.Correct3 < 2 Then
      Yellow_Room3.FeedbackGiven3 = True
    ElseIf Yellow_Room3.Correct3 = 2 Then
      Yellow_Room3.FeedbackGiven2 = True
    End If
    Orange_Room32.FeedbackGiven = True
    Red_Room321.FeedbackGiven = True
    Red_Room321.Show()
  ElseIf PriorRoom = 322 Then
```

```
            If Yellow_Room3.Correct3 < 2 Then
               Yellow_Room3.FeedbackGiven3 = True
            ElseIf Yellow_Room3.Correct3 = 2 Then
               Yellow_Room3.FeedbackGiven2 = True
            End If
            Orange_Room32.FeedbackGiven = True
            Red_Room322.FeedbackGiven = True
            Red_Room322.Show()
         ElseIf PriorRoom = 323 Then
            If Yellow_Room3.Correct3 < 2 Then
               Yellow_Room3.FeedbackGiven3 = True
            ElseIf Yellow_Room3.Correct3 = 2 Then
               Yellow_Room3.FeedbackGiven2 = True
            End If
            Orange_Room32.FeedbackGiven = True
            Red_Room323.FeedbackGiven = True
            Red_Room323.Show()
         End If
      Else
         If PriorRoom = 111 Then
            Red_Room111.Show()
         ElseIf PriorRoom = 112 Then
            Red_Room112.Show()
         ElseIf PriorRoom = 113 Then
            Red_Room113.Show()
         ElseIf PriorRoom = 121 Then
            Red_Room121.Show()
         ElseIf PriorRoom = 122 Then
            Red_Room122.Show()
         ElseIf PriorRoom = 123 Then
            Red_Room123.Show()
         ElseIf PriorRoom = 211 Then
            Red_Room211.Show()
         ElseIf PriorRoom = 212 Then
            Red_Room212.Show()
         ElseIf PriorRoom = 213 Then
            Red_Room213.Show()
         ElseIf PriorRoom = 221 Then
            Red_Room221.Show()
         ElseIf PriorRoom = 222 Then
            Red_Room222.Show()
         ElseIf PriorRoom = 223 Then
            Red_Room223.Show()
         ElseIf PriorRoom = 311 Then
            Red_Room311.Show()
         ElseIf PriorRoom = 312 Then
            Red_Room312.Show()
         ElseIf PriorRoom = 313 Then
            Red_Room313.Show()
         ElseIf PriorRoom = 321 Then
            Red_Room321.Show()
         ElseIf PriorRoom = 322 Then
            Red_Room322.Show()
         ElseIf PriorRoom = 323 Then
            Red_Room323.Show()
         End If
      End If
   Else
      If PriorRoom = 111 Then
         Red_Room111.Show()
      ElseIf PriorRoom = 112 Then
         Red_Room112.Show()
      ElseIf PriorRoom = 113 Then
```

```
            Red_Room113.Show()
        ElseIf PriorRoom = 121 Then
            Red_Room121.Show()
        ElseIf PriorRoom = 122 Then
            Red_Room122.Show()
        ElseIf PriorRoom = 123 Then
            Red_Room123.Show()
        ElseIf PriorRoom = 211 Then
            Red_Room211.Show()
        ElseIf PriorRoom = 212 Then
            Red_Room212.Show()
        ElseIf PriorRoom = 213 Then
            Red_Room213.Show()
        ElseIf PriorRoom = 221 Then
            Red_Room221.Show()
        ElseIf PriorRoom = 222 Then
            Red_Room222.Show()
        ElseIf PriorRoom = 223 Then
            Red_Room223.Show()
        ElseIf PriorRoom = 311 Then
            Red_Room311.Show()
        ElseIf PriorRoom = 312 Then
            Red_Room312.Show()
        ElseIf PriorRoom = 313 Then
            Red_Room313.Show()
        ElseIf PriorRoom = 321 Then
            Red_Room321.Show()
        ElseIf PriorRoom = 322 Then
            Red_Room322.Show()
        ElseIf PriorRoom = 323 Then
            Red_Room323.Show()
        End If
    End If


  End Sub

  'MAKES TIMER COUNT DOWN
  Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer2.Tick
    ProgressBar1.Value = Progress
    If Progress < 100 Then
      Progress = Progress + Yellow_Room1.AddNumber
    ElseIf Progress = 100 Then
      ShowPriorRoom()
      Me.Hide()
      Timer2.Enabled = False
      ProgressBar1.Value = 0
      ProgressBar1.Visible = False
      Label2.Visible = False
      Progress = 0
      Button4.Enabled = True
    End If
  End Sub

End Class
```

B.13 Treasure_Room.vb

```
Public Class Treasure_Room

  Dim Period As Integer
```

```
    Dim Room As Single
    Dim Maze As Integer
    Dim Direction As Integer
    Dim Int As Integer
    Dim IntMillisecond As Integer
    Dim Number As Integer

    Private Sub Treasure_Room_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Period = ID.Period
        Maze = ID.Maze
        Room = 4
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & vbCrLf, True)

        If Period = 1 Then
            ID.CompletedMaze1 = ID.CompletedMaze1 + 1
        ElseIf Period = 2 Then
            ID.CompletedMaze2 = ID.CompletedMaze2 + 1
        End If

        If Period = 2 Then
            Button1.Text = "" & ID.Counter1Red & " Remaining"
            Button2.Text = "" & ID.Counter2Blue & " Remaining"
            Button3.Text = "" & ID.Counter3Yellow & " Remaining"
            Button4.Visible = False
            Button1.Visible = True
            Button2.Visible = True
            Button3.Visible = True
            Label2.Visible = True

            'Disables button if completed all in quota
            If ID.Counter1Red = 0 Then
                Button1.Enabled = False
                Button1.BackColor = Color.FromArgb(CType(255, Byte), CType(192, Byte), CType(192, Byte))
            End If
            If ID.Counter2Blue = 0 Then
                Button2.Enabled = False
                Button2.BackColor = Color.FromArgb(CType(192, Byte), CType(192, Byte), CType(255, Byte))
            End If
            If ID.Counter3Yellow = 0 Then
                Button3.Enabled = False
                Button3.BackColor = Color.FromArgb(CType(255, Byte), CType(255, Byte), CType(192, Byte))
            End If
            If ID.Counter1Red = 0 Then
                If ID.Counter2Blue = 0 Then
                    If ID.Counter3Yellow = 0 Then
                        Button4.Visible = True
                        Button1.Visible = False
                        Button2.Visible = False
                        Button3.Visible = False
                        Label2.Visible = False
                    End If
                End If
            End If

        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Direction = 5
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
```

```
        UnloadAllForms()

        If ID.IntMillisecond Mod 3 = 0 Then
            ID.Cue = 1
        ElseIf ID.IntMillisecond Mod 3 = 1 Then
            ID.Cue = 3
        Else
            ID.Cue = 2
        End If
        ID.Counter1Red = ID.Counter1Red - 1

        ID.Maze = ID.Maze + 1

        Me.Close()
        Yellow_Room1.Show()

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Direction = 5
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        UnloadAllForms()

        If ID.IntMillisecond Mod 3 = 0 Then
            ID.Cue = 2
        ElseIf ID.IntMillisecond Mod 3 = 1 Then
            ID.Cue = 1
        Else
            ID.Cue = 3
        End If
        ID.Counter2Blue = ID.Counter2Blue - 1

        ID.Maze = ID.Maze + 1

        Me.Close()
        Yellow_Room1.Show()

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Direction = 5
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        UnloadAllForms()

        If ID.IntMillisecond Mod 3 = 0 Then
            ID.Cue = 3
        ElseIf ID.IntMillisecond Mod 3 = 1 Then
            ID.Cue = 2
        Else
            ID.Cue = 1
        End If
        ID.Counter3Yellow = ID.Counter3Yellow - 1

        ID.Maze = ID.Maze + 1

        Me.Close()
        Yellow_Room1.Show()

    End Sub
```

```
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        Direction = 5
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", ID.ID & "," & ID.Feedback & "," & ID.Compensation
& "," & Period & "," & Maze & "," & ID.Cue & "," & Room & "," & Date.Now & "," & Date.Now.Millisecond & "," & Direction &
vbCrLf, True)
        UnloadAllForms()

        ID.Maze = ID.Maze + 1

        If Period = 1 Then
            Number = ID.Maze + 9
            If ID.Maze < 10 Then
                If Number Mod 9 = 1 Then
                    ID.Cue = 1
                ElseIf Number Mod 9 = 2 Then
                    ID.Cue = 1
                ElseIf Number Mod 9 = 3 Then
                    ID.Cue = 1
                ElseIf Number Mod 9 = 4 Then
                    ID.Cue = 2
                ElseIf Number Mod 9 = 5 Then
                    ID.Cue = 2
                ElseIf Number Mod 9 = 6 Then
                    ID.Cue = 2
                ElseIf Number Mod 9 = 7 Then
                    ID.Cue = 3
                ElseIf Number Mod 9 = 8 Then
                    ID.Cue = 3
                ElseIf Number Mod 9 = 0 Then
                    ID.Cue = 3
                End If
            Else
                If Number Mod 9 = 1 Then
                    ID.Cue = 1
                ElseIf Number Mod 9 = 2 Then
                    ID.Cue = 2
                ElseIf Number Mod 9 = 3 Then
                    ID.Cue = 3
                ElseIf Number Mod 9 = 4 Then
                    ID.Cue = 1
                ElseIf Number Mod 9 = 5 Then
                    ID.Cue = 2
                ElseIf Number Mod 9 = 6 Then
                    ID.Cue = 3
                ElseIf Number Mod 9 = 7 Then
                    ID.Cue = 1
                ElseIf Number Mod 9 = 8 Then
                    ID.Cue = 2
                ElseIf Number Mod 9 = 0 Then
                    ID.Cue = 3
                End If
            End If
        Else
            IntMillisecond = Date.Now.Millisecond
            'Determines random presentation of maze/cue combinations
            Int = CInt(Rnd() * IntMillisecond)
            If Int Mod 3 = 1 Then
                ID.Cue = 2
            ElseIf Int Mod 3 = 2 Then
                ID.Cue = 3
            Else
                ID.Cue = 1
```

```
            End If
        End If

        Me.Close()
        Yellow_Room1.Show()

        'End game if too many mazes completed
        If Period = 1 Then
            If ID.CompletedMaze1 = ID.Max1 Then
                ID.PaidTime1 = ID.RemainingTime
                ID.Minutes1 = ID.PaidTime1.Minutes
                ID.Seconds1 = ID.PaidTime1.Seconds
                ID.TotalSeconds1 = (ID.Minutes1 * 60) + ID.Seconds1
                MsgBox("Congratulations!  You completed all of the available mazes in the first round with " & ID.TotalSeconds1 & "
seconds remaining.  Click "OK" to see your compensation.")
                ID.FinishedEarly1 = True
                ID.EndingProcedure()
            End If
        ElseIf Period = 2 Then
            If ID.CompletedMaze2 = ID.Max2 Then
                ID.PaidTime2 = ID.RemainingTime
                ID.Minutes2 = ID.PaidTime2.Minutes
                ID.Seconds2 = ID.PaidTime2.Seconds
                ID.TotalSeconds2 = (ID.Minutes2 * 60) + ID.Seconds2
                MsgBox("Congratulations!  You completed all of the available mazes in the second round with " & ID.TotalSeconds2 & "
seconds remaining.  Click "OK" to see your compensation.")
                ID.FinishedEarly2 = True
                ID.EndingProcedure()
            End If
        End If
    End Sub

End Class
```

B.14 dlgIntermission.vb

```
Public Class dlgIntermission
    Dim Fixed As String
    Dim Rate As String
    Dim SecondRate As String

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If ID.IntMillisecond Mod 3 = 0 Then
            ID.Cue = 1
        ElseIf ID.IntMillisecond Mod 3 = 1 Then
            ID.Cue = 3
        Else
            ID.Cue = 2
        End If
        ID.Counter1Red = ID.Counter1Red - 1
        ID.AlarmTime = DateAdd(DateInterval.Minute, ID.Min2, Date.Now)
        Yellow_Room1.Show()
        ID.Timer1.Enabled = True
        Me.Close()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        If ID.IntMillisecond Mod 3 = 0 Then
            ID.Cue = 2
        ElseIf ID.IntMillisecond Mod 3 = 1 Then
            ID.Cue = 1
        Else
```

```
            ID.Cue = 3
        End If
        ID.Counter2Blue = ID.Counter2Blue - 1
        ID.AlarmTime = DateAdd(DateInterval.Minute, ID.Min2, Date.Now)
        Yellow_Room1.Show()
        ID.Timer1.Enabled = True
        Me.Close()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        If ID.IntMillisecond Mod 3 = 0 Then
            ID.Cue = 3
        ElseIf ID.IntMillisecond Mod 3 = 1 Then
            ID.Cue = 2
        Else
            ID.Cue = 1
        End If
        ID.Counter3Yellow = ID.Counter3Yellow - 1
        ID.AlarmTime = DateAdd(DateInterval.Minute, ID.Min2, Date.Now)
        Yellow_Room1.Show()
        ID.Timer1.Enabled = True
        Me.Close()
    End Sub

    Private Sub dlgIntermission_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        ID.Period = 2
        Fixed = FormatCurrency((2 * ID.Fixed), 2)
        Rate = FormatCurrency((ID.Rate), 2)
        SecondRate = FormatCurrency((ID.SecondRate), 2)
        If ID.Compensation = 1 Then
            RichTextBox1.Text = "In the second round, you will have " & ID.Min2 & " minutes to work on the maze task. In this round, you will NOT be notified that you have entered an incorrect path." & vbCrLf & vbCrLf & "Please remember that the same color-cue pattern combinations from the mazes in the first round will also be used in the second round." & vbCrLf & vbCrLf & "The number of mazes you complete is highly valued." & vbCrLf & vbCrLf & "You will be paid " & Rate & " for each maze you complete in this round (up to a maximum of " & ID.Max2 & ").  In the event that you complete all of the allotted mazes for the second round, you will be paid a bonus of " & SecondRate & " for each second remaining." & vbCrLf & vbCrLf & "Please click on the button below that matches the door color of the maze you would like to work on (this will initiate the second round)."
        Else
            RichTextBox1.Text = "In the second round, you will have " & ID.Min2 & " minutes to work on the maze task. In this round, you will NOT be notified that you have entered an incorrect path." & vbCrLf & vbCrLf & "Please remember that the same color-cue pattern combinations from the mazes in the first round will also be used in the second round." & vbCrLf & vbCrLf & "The number of mazes you complete is highly valued." & vbCrLf & vbCrLf & "You will be paid " & Fixed & " for working on the maze task in this round, regardless of the number of mazes you complete (up to a maximum of " & ID.Max2 & ").  In the event that you complete all of the allotted mazes for the second round, you will be paid a bonus of " & SecondRate & " for each second remaining." & vbCrLf & vbCrLf & "Please click on the button below that matches the door color of the maze you would like to work on (this will initiate the second round)."
        End If
        Button1.Text = "" & ID.Counter1Red & " Remaining"
        Button2.Text = "" & ID.Counter2Blue & " Remaining"
        Button3.Text = "" & ID.Counter3Yellow & " Remaining"
    End Sub
End Class
```

B.15 QuestFinal.vb

```
Public Class QuestFinal
    Public nOne As Integer
    Public nTwo As Integer
    Public nThree As Integer
    Public nFour As Integer
    Public nFive As Integer
    Public nSix_1 As Single
```

```
Public nSix_2 As Single
Public nSeven As String
Public Gender As String
Public nYear As Integer
Dim Indicator7 As Boolean
Dim nInd As Integer
Dim IndicatorG As Boolean
Dim IndicatorA As Boolean
Public nAge As Single

Private Sub btnSubmit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubmit.Click
   If rb1_n4.Checked = False Then
      If rb1_n3.Checked = False Then
         If rb1_n2.Checked = False Then
            If rb1_n1.Checked = False Then
               If rb1_0.Checked = False Then
                  If rb1_1.Checked = False Then
                     If rb1_2.Checked = False Then
                        If rb1_3.Checked = False Then
                           If rb1_4.Checked = False Then
                              MsgBox("Please select a response for question 1.  Click OK to continue.")
                              nOne = -5
                           End If
                        End If
                     End If
                  End If
               End If
            End If
         End If
      End If
   End If
   If rb1_n4.Checked = True Then
      nOne = -4
   ElseIf rb1_n3.Checked = True Then
      nOne = -3
   ElseIf rb1_n2.Checked = True Then
      nOne = -2
   ElseIf rb1_n1.Checked = True Then
      nOne = -1
   ElseIf rb1_0.Checked = True Then
      nOne = 0
   ElseIf rb1_1.Checked = True Then
      nOne = 1
   ElseIf rb1_2.Checked = True Then
      nOne = 2
   ElseIf rb1_3.Checked = True Then
      nOne = 3
   ElseIf rb1_4.Checked = True Then
      nOne = 4
   End If
   If rb2_n4.Checked = False Then
      If rb2_n3.Checked = False Then
         If rb2_n2.Checked = False Then
            If rb2_n1.Checked = False Then
               If rb2_0.Checked = False Then
                  If rb2_1.Checked = False Then
                     If rb2_2.Checked = False Then
                        If rb2_3.Checked = False Then
                           If rb2_4.Checked = False Then
                              MsgBox("Please select a response for question 2.  Click OK to continue.")
                              nTwo = -5
                           End If
                        End If
```

```
                End If
            End If
        End If
    End If
End If
End If
If rb2_n4.Checked = True Then
    nTwo = -4
ElseIf rb2_n3.Checked = True Then
    nTwo = -3
ElseIf rb2_n2.Checked = True Then
    nTwo = -2
ElseIf rb2_n1.Checked = True Then
    nTwo = -1
ElseIf rb2_0.Checked = True Then
    nTwo = 0
ElseIf rb2_1.Checked = True Then
    nTwo = 1
ElseIf rb2_2.Checked = True Then
    nTwo = 2
ElseIf rb2_3.Checked = True Then
    nTwo = 3
ElseIf rb2_4.Checked = True Then
    nTwo = 4
End If
If rb3_n4.Checked = False Then
    If rb3_n3.Checked = False Then
        If rb3_n2.Checked = False Then
            If rb3_n1.Checked = False Then
                If rb3_0.Checked = False Then
                    If rb3_1.Checked = False Then
                        If rb3_2.Checked = False Then
                            If rb3_3.Checked = False Then
                                If rb3_4.Checked = False Then
                                    MsgBox("Please select a response for question 3.  Click OK to continue.")
                                    nThree = -5
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
End If
If rb3_n4.Checked = True Then
    nThree = -4
ElseIf rb3_n3.Checked = True Then
    nThree = -3
ElseIf rb3_n2.Checked = True Then
    nThree = -2
ElseIf rb3_n1.Checked = True Then
    nThree = -1
ElseIf rb3_0.Checked = True Then
    nThree = 0
ElseIf rb3_1.Checked = True Then
    nThree = 1
ElseIf rb3_2.Checked = True Then
    nThree = 2
ElseIf rb3_3.Checked = True Then
    nThree = 3
ElseIf rb3_4.Checked = True Then
```

```
    nThree = 4
End If
If rb4_n4.Checked = False Then
   If rb4_n3.Checked = False Then
      If rb4_n2.Checked = False Then
         If rb4_n1.Checked = False Then
            If rb4_0.Checked = False Then
               If rb4_1.Checked = False Then
                  If rb4_2.Checked = False Then
                     If rb4_3.Checked = False Then
                        If rb4_4.Checked = False Then
                           MsgBox("Please select a response for question 4.  Click OK to continue.")
                           nFour = -5
                        End If
                     End If
                  End If
               End If
            End If
         End If
      End If
   End If
End If
If rb4_n4.Checked = True Then
   nFour = -4
ElseIf rb4_n3.Checked = True Then
   nFour = -3
ElseIf rb4_n2.Checked = True Then
   nFour = -2
ElseIf rb4_n1.Checked = True Then
   nFour = -1
ElseIf rb4_0.Checked = True Then
   nFour = 0
ElseIf rb4_1.Checked = True Then
   nFour = 1
ElseIf rb4_2.Checked = True Then
   nFour = 2
ElseIf rb4_3.Checked = True Then
   nFour = 3
ElseIf rb4_4.Checked = True Then
   nFour = 4
End If
If rb5_n4.Checked = False Then
   If rb5_n3.Checked = False Then
      If rb5_n2.Checked = False Then
         If rb5_n1.Checked = False Then
            If rb5_0.Checked = False Then
               If rb5_1.Checked = False Then
                  If rb5_2.Checked = False Then
                     If rb5_3.Checked = False Then
                        If rb5_4.Checked = False Then
                           MsgBox("Please select a response for question 5.  Click OK to continue.")
                           nFive = -5
                        End If
                     End If
                  End If
               End If
            End If
         End If
      End If
   End If
End If
If rb5_n4.Checked = True Then
   nFive = -4
```

```
    ElseIf rb5_n3.Checked = True Then
        nFive = -3
    ElseIf rb5_n2.Checked = True Then
        nFive = -2
    ElseIf rb5_n1.Checked = True Then
        nFive = -1
    ElseIf rb5_0.Checked = True Then
        nFive = 0
    ElseIf rb5_1.Checked = True Then
        nFive = 1
    ElseIf rb5_2.Checked = True Then
        nFive = 2
    ElseIf rb5_3.Checked = True Then
        nFive = 3
    ElseIf rb5_4.Checked = True Then
        nFive = 4
    End If
    If txt6_1.Text = "" Then
        MsgBox("Please enter the percentage of time you spent in Round 1 (0 to 100) looking for cues and patterns.  Click OK to
continue.")
        txt6_1.Select()
        nSix_1 = -1
    ElseIf Not IsNumeric(txt6_1.Text) Then
        MsgBox("Please enter a numeric percentage (excluding the % sign) for Round 1.  Click OK to continue.")
        txt6_1.Text = ""
        txt6_1.Select()
        nSix_1 = -1
    ElseIf Val(txt6_1.Text) < 0 Then
        MsgBox("Please enter a valid percentage of time for Round 1 (0 to 100).  Click OK to continue.")
        txt6_1.Text = ""
        txt6_1.Select()
        nSix_1 = -1
    ElseIf Val(txt6_1.Text) > 100 Then
        MsgBox("Please enter a valid percentage of time for Round 1 (0 to 100).  Click OK to continue.")
        txt6_1.Text = ""
        txt6_1.Select()
        nSix_1 = -1
    Else
        nSix_1 = Val(txt6_1.Text)
    End If
    If txt6_2.Text = "" Then
        MsgBox("Please enter the percentage of time you spent in Round 2 (0 to 100) looking for cues and patterns.  Click OK to
continue.")
        txt6_2.Select()
        nSix_2 = -1
    ElseIf Not IsNumeric(txt6_2.Text) Then
        MsgBox("Please enter a numeric percentage (excluding the % sign) for Round 2.  Click OK to continue.")
        txt6_2.Text = ""
        txt6_2.Select()
        nSix_2 = -1
    ElseIf Val(txt6_2.Text) < 0 Then
        MsgBox("Please enter a valid percentage of time for Round 2 (0 to 100).  Click OK to continue.")
        txt6_2.Text = ""
        txt6_2.Select()
        nSix_2 = -1
    ElseIf Val(txt6_2.Text) > 100 Then
        MsgBox("Please enter a valid percentage of time for Round 2 (0 to 100).  Click OK to continue.")
        txt6_2.Text = ""
        txt6_2.Select()
        nSix_2 = -1
    Else
        nSix_2 = Val(txt6_2.Text)
    End If
```

```
    If nInd <> 1 Then
      If txt7.Text <> "" Then
        Indicator7 = True
        nSeven = txt7.Text
      Else
        MsgBox("If you have any additional information that might be helpful for this study, please write them in the box for
question 6.")
        txt7.Select()
        nInd = 1
      End If
    Else
      Indicator7 = True
      nSeven = txt7.Text
    End If
    If rbMale.Checked = False Then
      If rbFemale.Checked = False Then
        MsgBox("Please select a gender in Background Information.  Click OK to continue.")
      End If
    End If
    If rbMale.Checked = True Then
      Gender = "Male"
      IndicatorG = True
    ElseIf rbFemale.Checked = True Then
      Gender = "Female"
      IndicatorG = True
    End If
    If txtAge.Text = "" Then
      MsgBox("Please enter your age in the Background Information.  Click OK to continue.")
      txtAge.Select()
    Else
      If Not IsNumeric(txtAge.Text) Then
        MsgBox("Please enter a numeric age in the Background Information.  Click OK to continue.")
        txtAge.Text = ""
        txtAge.Select()
      ElseIf Val(txtAge.Text) < 12 Then
        MsgBox("Please enter a valid age (12 or greater).  Click OK to continue.")
        txtAge.Text = ""
        txtAge.Select()
      Else
        nAge = Val(txtAge.Text)
        IndicatorA = True
      End If
    End If
    If rbY1.Checked = False Then
      If rbY2.Checked = False Then
        If rbY3.Checked = False Then
          If rbY4.Checked = False Then
            If rbY5plus.Checked = False Then
              MsgBox("Please select a year in school in Background Information.  Click OK to continue.")
            End If
          End If
        End If
      End If
    End If
    If rbY1.Checked = True Then
      nYear = 1
    ElseIf rbY2.Checked = True Then
      nYear = 2
    ElseIf rbY3.Checked = True Then
      nYear = 3
    ElseIf rbY4.Checked = True Then
      nYear = 4
    ElseIf rbY5plus.Checked = True Then
```

```
            nYear = 5
        End If
        If nOne > -5 Then
            If nTwo > -5 Then
                If nThree > -5 Then
                    If nFour > -5 Then
                        If nFive > -5 Then
                            If nSix_1 > -1 Then
                                If nSix_2 > -1 Then
                                    If Indicator7 = True Then
                                        If IndicatorG = True Then
                                            If IndicatorA = True Then
                                                If nYear > 0 Then
                                                    Final.Show()
                                                    Me.Hide()
                                                End If
                                            End If
                                        End If
                                    End If
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End Sub
    Private Sub frmQuestFinal_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        MsgBox("Please take a few minutes to answer these final questions about your reactions to the task you just completed and about
your background. Click OK to continue.", , "")
        rb1_n4.Checked = False
        rb1_n4.TabStop = True
    End Sub
End Class
```

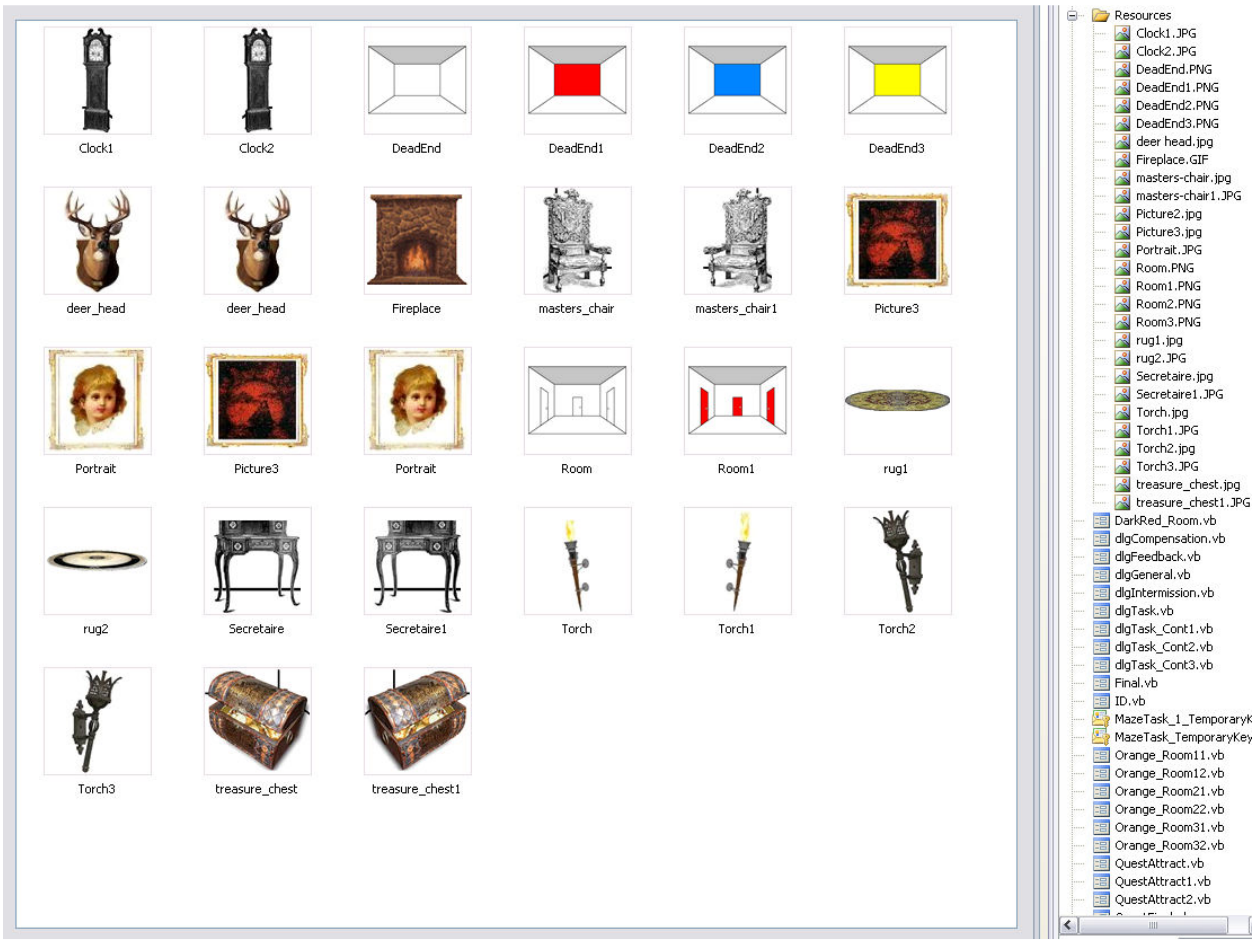
B.16 Final.vb

```
Public Class Final
    Dim CompTotal As String

    Private Sub frmFinal_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        CompTotal = FormatCurrency((ID.Comp1 + ID.Comp2), 2)
        lblCorrect1.Text = ID.CompletedMaze1
        lblComp1.Text = ID.CashComp1
        lblCorrect2.Text = ID.CompletedMaze2
        lblComp2.Text = ID.CashComp2
        lblCompTotal.Text = CompTotal
        StoreData()
    End Sub

    Public Sub StoreData()
        My.Computer.FileSystem.WriteAllText("Participant_" & ID.ID & ".csv", GetOutputString(), True)
    End Sub

    Private Function GetOutputString() As String
        Dim str As String = ""

        str += "" & vbCrLf
        str += "Cue 1 Color, Cue 2 Color, Cue 3 Color" & vbCrLf
        str += ID.Color1 & "," & ID.Color2 & "," & ID.Color3 & vbCrLf
        str += "" & vbCrLf
```

```
        str += "# Completed (Round 1)," & ID.CompletedMaze1 & vbCrLf
        str += "Remaining Seconds (Round 1)," & ID.TotalSeconds1 & vbCrLf
        str += "Compensation (Round 1)," & ID.CashComp1 & vbCrLf
        str += "" & vbCrLf
        str += "# Completed (Round 2)," & ID.CompletedMaze2 & vbCrLf
        str += "Remaining Seconds (Round 2)," & ID.TotalSeconds2 & vbCrLf
        str += "Compensation (Round 2)," & ID.CashComp2 & vbCrLf
        str += "" & vbCrLf
        str += "Total Compensation," & CompTotal & vbCrLf
        str += "" & vbCrLf

        str += "First Task Attractiveness Questionnaire" & vbCrLf
        str += "Second Task Attractiveness Questionnaire" & vbCrLf
        str += "Third Task Attractiveness Questionnaire" & vbCrLf
        str += "" & vbCrLf

        str += "Attractive/Repulsive," & QuestAttract.nOne & vbCrLf
        str += "Exciting/Dull," & QuestAttract.nTwo & vbCrLf
        str += "Good/Bad," & QuestAttract.nThree & vbCrLf
        str += "Interesting/Boring," & QuestAttract.nFour & vbCrLf
        str += "Superior/Inferior," & QuestAttract.nFive & vbCrLf
        str += "Wholesome/Unwholesome," & QuestAttract.nSix & vbCrLf
        str += "Fun/Tedious," & QuestAttract.nSeven & vbCrLf
        str += "Attractive/Repulsive," & QuestAttract1.nOne & vbCrLf
        str += "Exciting/Dull," & QuestAttract1.nTwo & vbCrLf
        str += "Good/Bad," & QuestAttract1.nThree & vbCrLf
        str += "Interesting/Boring," & QuestAttract1.nFour & vbCrLf
        str += "Superior/Inferior," & QuestAttract1.nFive & vbCrLf
        str += "Wholesome/Unwholesome," & QuestAttract1.nSix & vbCrLf
        str += "Fun/Tedious," & QuestAttract1.nSeven & vbCrLf
        str += "Attractive/Repulsive," & QuestAttract2.nOne & vbCrLf
        str += "Exciting/Dull," & QuestAttract2.nTwo & vbCrLf
        str += "Good/Bad," & QuestAttract2.nThree & vbCrLf
        str += "Interesting/Boring," & QuestAttract2.nFour & vbCrLf
        str += "Superior/Inferior," & QuestAttract2.nFive & vbCrLf
        str += "Wholesome/Unwholesome," & QuestAttract2.nSix & vbCrLf
        str += "Fun/Tedious," & QuestAttract2.nSeven & vbCrLf
        str += "" & vbCrLf

        str += "Final Questionnaire" & vbCrLf
        str += "Question 1," & QuestFinal.nOne & vbCrLf
        str += "Question 2," & QuestFinal.nTwo & vbCrLf
        str += "Question 3," & QuestFinal.nThree & vbCrLf
        str += "Question 4," & QuestFinal.nFour & vbCrLf
        str += "Question 5," & QuestFinal.nFive & vbCrLf
        str += "Question 6 (Round 1)," & QuestFinal.nSix_1 & vbCrLf
        str += "Question 6 (Round 2)," & QuestFinal.nSix_2 & vbCrLf
        str += "Gender," & QuestFinal.Gender & vbCrLf
        str += "Age," & QuestFinal.nAge & vbCrLf
        str += "Year in school," & QuestFinal.nYear & vbCrLf
        str += "Question 7," & QuestFinal.nSeven & vbCrLf
        str += "" & vbCrLf

        Return str
    End Function

    Private Sub Label3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label3.Click
        ID.Close()
        QuestAttract.Close()
        QuestAttract1.Close()
        QuestAttract2.Close()
        QuestFinal.Close()
        Me.Close()
```

```
    End Sub
End Class
```

B.17 Resources



Bibliography

Anderhub, V., W. Güth, U. Gneezy, and D. Sonsino. 2001. On the interaction of risk and time preferences: an experimental study. *German Economic Review* 2(3): 239–253.

Balakrishnan, R., K. Sivaramakrishnan, and G. B. Sprinkle. 2009. *Managerial Accounting*. Hoboken, NJ: John Wiley & Sons, Inc.

Balzer, W. K., M. E. Doherty, and R. O'Conner, Jr. 1989. Effects of cognitive feedback on performance. *Psychological Bulletin* 106(3): 410–433.

Bonner, S. E., and G. B. Sprinkle. 2002. The effects of monetary incentives on effort and task performance: theories, evidence, and a framework for research. *Accounting, Organizations and Society* 27: 303–345.

Bonner, S. E. and P. L. Walker. 1994. The effects of instruction and experience on the acquisition of auditing knowledge. *The Accounting Review* 69(1): 157–178.

Brehmer, B. 1995. Feedback delays in complex dynamic decision tasks. In P. French & J. Funke (Eds.) *Complex Problem Solving: The European Perspective*. Hillsdale, NJ: Erlbaum.

Brewster, B. E. 2011. How a systems perspective improves knowledge acquisition and performance in analytical procedures. *The Accounting Review* 86(3): 915–943.

Butler, A. C., J. D. Karpicke, and H. L. Roediger III. 2007. The effect of type and timing of feedback on learning from multiple-choice tests. *Journal of Experimental Psychology: Applied* 13(4): 273–281.

Casey, Jr. C. J. 1980. Variation in accounting information load: the effect on loan officers' predictions of bankruptcy. *The Accounting Review* 55(1): 36–49.

Conover, W. J., and R. L. Iman. 1981. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician* 35(3): 124–129.

Diehl, E., and J. D. Sterman. 1995. Effects of feedback complexity on dynamic decision making. *Organizational Behavior and Human Decision Processes* 62(2): 198–215.

Einhorn, H. J. 1980. Learning from experience and suboptimal rules in decision making. In T. S. Wallsten (Ed.) *Cognitive Processes in Choice and Decision Behavior*. Hillsdale, New Jersey: Erlbaum.

Einhorn, H. J., and R. M. Hogarth. 1981. Behavioral decision theory: processes of judgment and choice. *Journal of Accounting Research* 19(1): 1–31.

Frederickson, J. R., S. A. Peffer, and J. Pratt. 1999. Performance evaluation judgments: effects of prior experience under different performance evaluation schemes and feedback frequencies. *Journal of Accounting Research* 37(1): 151–165.

Friedman, M. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32: 675–701.

Gabaix, X., and D. Laibson. 2005. Bounded rationality and directed cognition. Working Paper, Massachusetts Institute of Technology, Harvard University, and NBER.

Gabaix, X., D. Laibson, G. Moloche, and S. Weinberg. 2006. Costly information acquisition: experimental analysis of a boundedly rational model. *American Economic Review* 96(4): 1043–1068.

Gibson, F. P. 2000. Feedback delays: how can decision makers learn not to buy a new car every time the garage is empty? *Organizational Behavior and Human Decision Processes* 83(1): 141–166.

Hales, J., and M. G. Williamson. 2010. Implicit employment contracts: the limits of management reputation for promoting firm productivity. *Journal of Accounting Research* 48(1): 147–176.

Hannan, R. L., R. Krishnan, and A. H. Newman. 2008. The effects of disseminating relative performance feedback in tournament and individual performance compensation schemes. *The Accounting Review* 83(4): 893–913.

Hertwig, R., and P. M. Todd. 2003. More is not always better: the benefits of cognitive limits. In D. Hardman and L. Macci (Editors), *Thinking: Psychological Perspectives on Reasoning, Judgment and Decision Making:* 212–231. Chichester, England: Wiley.

Iselin, E. R. 1988. The effects of information load and information diversity on decision quality in a structured decision task. *Accounting, Organizations and Society* 13(2): 147–164.

Jacoby, J., D. E. Speller, C. K. Berning. 1974. Brand choice behavior as a function of information load: replication and extension. *The Journal of Consumer Research* 1(1): 33–42.

Jensen, M. C. 2001. *Foundations for Organizational Strategy*. Cambridge, MA: Harvard University Press.

Kachelmeier, S. J., and M. H. Granof. 1993. Depreciation and capital investment decisions: experimental evidence in a governmental setting. *Journal of Accounting and Public Policy* 12: 291–323.

Kahneman, D., and A. Tversky. 1979. Prospect theory: an analysis of decision under risk. *Econometrica* 47: 263–290.

Keren, G., and P. Roelofsma. 1995. Immediacy and certainty in intertemporal choice. *Organizational Behavior and Human Decision Processes* 63(3): 287–297.

Kirk, R. E. 1995. *Experimental Design: Procedures for the Behavioral Sciences*. Pacific Grove, CA: Brooks/Cole Publishing Company.

Kirsch, L. J. 1996. The management of complex tasks in organizations: controlling the systems development process. *Organization Science* 7(1): 1–21.

Kulik, J. A., and C.-L. C. Kulik. 1988. Timing of feedback and verbal learning. *Review of Educational Research* 58(1): 79–97.

Lewis, M. W., and J. R. Anderson. 1985. Discrimination of operator schemata in problem solving: learning from examples. *Cognitive Psychology* 17: 26–65.

Libby, R., and J. Luft. 1993. Determinants of judgment performance in accounting setting: ability, knowledge, motivation, and environment. *Accounting, Organizations and Society* 18(5): 425–450.

Loewenstein, G., and D. Prelec. 1992. Anomalies in intertemporal choice: evidence and an interpretation. *The Quarterly Journal of Economics* 107(2): 573–597.

Luckett, P. F., and I. R. C. Eggleton. 1991. Feedback and management accounting: a review of research into behavioral consequences. *Accounting, Organizations and Society* 16(4): 371–394.

Luft, J., and M. D. Shields. 2010. Psychology models in management accounting. *Foundations and Trends in Accounting* 4(3–4): 199–345.

Miller, G. A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 63: 81–97.

Peecher, M. E., R. Schwartz, and I. Solomon. 2007. It's all about quality: perspectives on strategic-systems auditing. *Accounting, Organizations and Society* 32: 463–485.

Ray, K. 2007. The retention effect of withholding performance information. *The Accounting Review* 82(2): 389–425.

Rosenthal, R., and R. L. Rosnow. 1991. *Essentials of Behavioral Research: Methods and Data Analysis*. New York, NY: McGraw-Hill

Salmoni, A. W., R. A. Schmidt., and C. B. Walter. 1984. Knowledge of results and motor learning: a review and critical reappraisal. *Psychological Bulletin* 95(3): 355–386.

Schmidt, R. A. 1991. Frequent augmented feedback can degrade learning: evidence and interpretation. In J. Requin & G. E. Stelmach (Eds.) *Tutorials in Motor Science*: 59–75. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Schmidt, R. A., D. E. Young, S. Swinnen, and D. C. Shaprio. 1989. Summary knowledge of results for skill acquisition: support for the guidance hypothesis. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 15(2): 352–359.

Shields, M. D. 1983. Effects of information supply and demand on judgment accuracy: evidence from corporate managers. *The Accounting Review* 58(2): 284–303.

Simon, H. A. 1973. The structure of ill structure problems. *Artificial Intelligence* 4(3-4): 181–201.

Sprinkle, G. B. 2000. The effect of incentive contracts on learning and performance. *The Accounting Review* 75(3): 299–326.

Sprinkle, G. B., and M. G. Williamson. 2007. Experimental research in managerial accounting. *Handbook of Management Accounting Research* 17: 415–444.

Thaler, R. H., and H. M. Shefrin. 1981. An economic theory of self-control. *The Journal of Political Economy* 89(2): 392–406.

Tuttle, B., and F. G. Burton. 1999. The effects of a modest incentive on information overload in an investment analysis task. *Accounting, Organizations & Society* 24: 673–687.

Vita

Todd Ammon Thornock is the son of Warren Eldon Thornock and Sue Eileen Smith. He is married to Samara Elizabeth Fuller. He was born on June 14, 1977 in Sacramento, California. After graduating from Elk Grove High School in Elk Grove, California, in 1995, he attended Ricks College in Rexburg, Idaho. He graduated with an Associate degree of General Studies in the Honors Program in August of 1996. Todd then served a mission for the Church of Jesus Christ of Latter-day Saints in Concepción, Chile from September 1996 to August 1998. He then attended Brigham Young University and studied accounting, with an emphasis on professional accounting. In April of 2002, he received a Bachelor of Science degree in Accounting, cum laude with a minor in Spanish. Also in April of 2002, he received a Masters of Accounting from Brigham Young University with a minor in Information Systems. After graduation, Todd worked for PricewaterhouseCoopers, LLP in San José, California and Austin, Texas as a junior and senior associate in the audit practice. In August of 2006, Todd entered the Graduate School at the University of Texas at Austin. In August of 2009, he received a Master of Science in Accounting from the Graduate School. Todd's research examines the effects of incentives and feedback systems on individual performance and learning.

Permanent Address: 3120 Weston Drive, Ames, IA 50010

This dissertation was typed by the author.